AN ARGUS SPECIALIST PUBLICATION

November 1985

£1.00

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

Now Incorporating...

YOUR 64



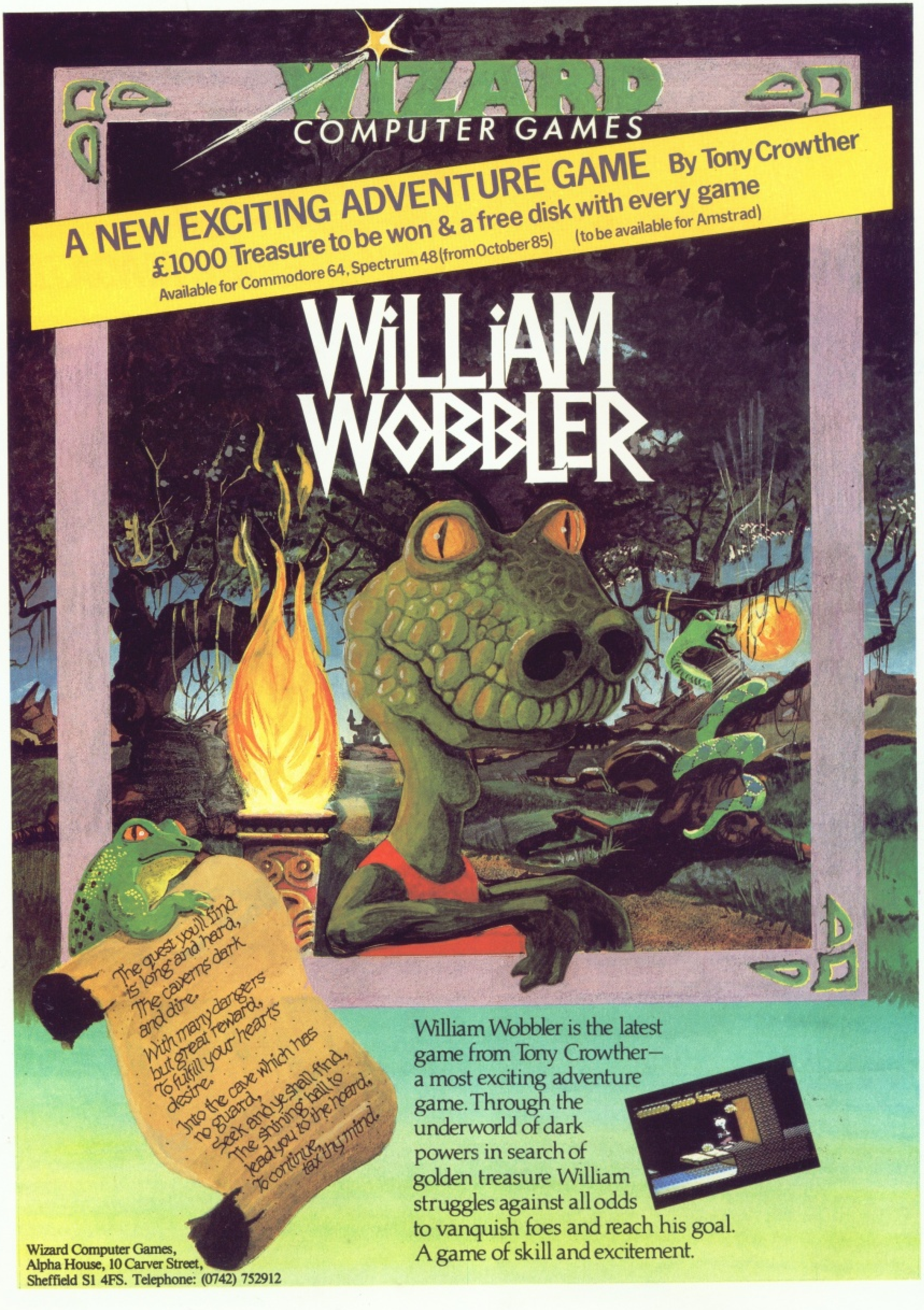
Mach 1— the first part of our type in assembler

Inside the 128

Full colour Action Replay

Competition— 10 Commodore modems must be won

Summer Games II— Game of the month



WIZARD
COMPUTER GAMES
A NEW EXCITING ADVENTURE GAME By Tony Crowther
£1000 Treasure to be won & a free disk with every game
Available for Commodore 64, Spectrum 48 (from October 85) (to be available for Amstrad)
WILLIAM WOBBLER
The quest you'll find is long and hard, The caverns dark and dire. With many dangers but great reward, To fulfill your hearts desire. Into the cave which has no guard, Seek and ye shall find, The shining ball to lead you to the hoard, To continue, tax thy mind.
William Wobbler is the latest game from Tony Crowther— a most exciting adventure game. Through the underworld of dark powers in search of golden treasure William struggles against all odds to vanquish foes and reach his goal. A game of skill and excitement.
Wizard Computer Games, Alpha House, 10 Carver Street, Sheffield S1 4FS. Telephone: (0742) 752912

VOLUME 2 NUMBER 2 NOVEMBER 1985

CONTENTS

FEATURES

SERIES

REGULARS

GAMES AND UTILITIES

COMPETITION

Editor: Stuart Cooke

Assistant Editor: Marie Curry

Advertisement Manager: Mike Segrue

Advertisement Copy Control: Laura Champion

Group Editor: Dave Bradshaw

Group Managing Editor: Wendy Palmer

Managing Director: Peter Welham

Origination: Ebony Typesetting

Design: Argus Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB.
Telephone: 01-437 0626
Telex: 8811896

Your Commodore is a monthly magazine appearing on the first Friday of each month.

Distribution by: Argus Press Sales & Distribution Ltd. 12-18 Paul Street, London EC2A 4JS. Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1985 Argus Specialist Publications Limited

Our COMMENT

THIS MONTH, WE'RE DELIGHTED TO announce that Your Commodore has now joined forces with Your 64 to bring you the best Commodore magazine around.

Publishers of Your Commodore, Argus Specialist Publications, recently concluded the purchase of Your 64 from Sportscene Press. Peter Welham, ASP's MD, said: "This acquisition consolidates our position as one of the leading titles in the buoyant Commodore market and further confirms our commitment to machine specific titles."

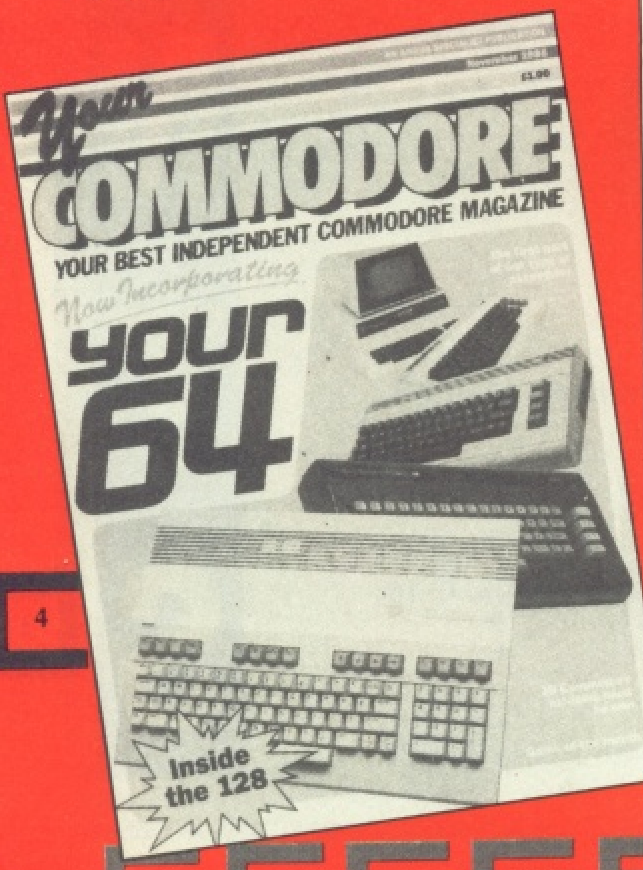
We're pleased to welcome all our new readers to our pages and we can promise that you won't be disappointed. we're really going to have our work cut out over the next few months making sure we bring you only the best in software and hardware reviews, games and utility listings, plus many exciting and absorbing features for you to relish.

We're also very conscious of the fact that a magazine is nothing without its readers and so we'll be waiting anxiously for the postman every day to find out what you really think about our magazine.

Write to us at Your Commodore, No 1 Golden Square, London W1R 3AB and tell us what you think are the best (and worst!) features of the magazine and give us your ideas on what needs to be included. We promise to read everything you send so get those letters in the post now.

Stay with us over the next few months and we'll bring you the only Commodore magazine worth reading.

Stuart



DATA STATEMENTS



Part of Ariolasoft's new range

The sky's the limit

ARIOLASOFT'S RECENT SUCCESS, SKYfox, has now been released on cassette priced at £9.95. The game was originally only available on disk and the cassette version eventually released was the third attempt at maintaining the standard of Skyfox since two earlier conversions were rejected by the manufacturers as being not up to scratch.

Skyfox is only one of a number of new releases from Ariolasoft. The latest titles are Racing Destruction Set, Pinball Construction Set, Adventure Construction Set, Music Construction Set, Seven Cities of Gold and Mail Order Monsters. So far these titles are only available on disk and are all priced at £14.95.

Ariolasoft can be contacted at Suite 105/105 Asphalte House, Palace St, London SW1E 5HS.

Chatting up

COMPUNET IS DEVELOPING A NEW SERvice scheduled for a late autumn launch. It is a scrolling, CB style chat facility. The chat line will allow users to hold conversations in real time with other users all over the country.

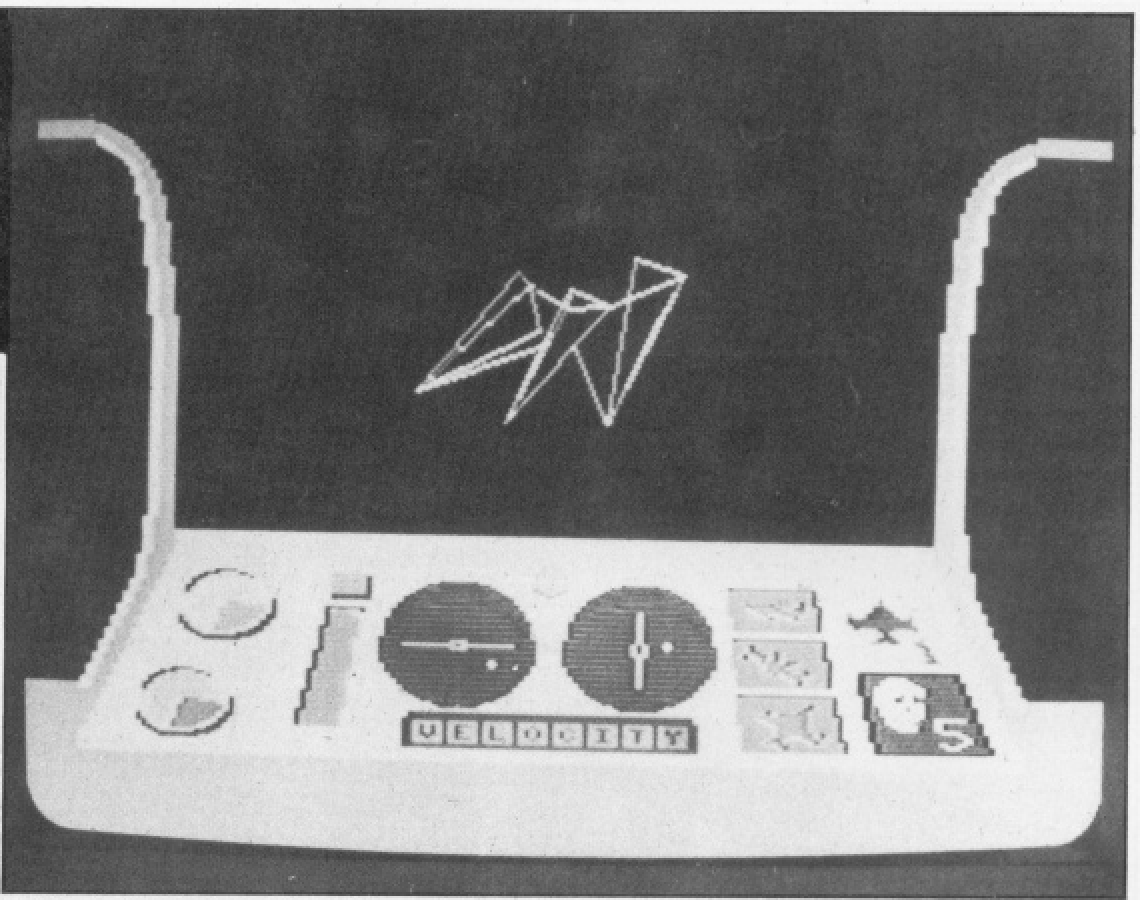
Contributions are typed into a window at the bottom of the screen while a second window above displays messages from other users. By scrolling back and forth the whole conversation can be reviewed. Users with similar interests can form groups by using individual chat lines which will be provided and those wanting a chat can monitor various conversations before deciding which one to join.

For further information contact: Compunet, Metford House, 15-18 Clipstone St, London W1P 7DF.

Starstruck

STARION FROM MELBOURNE HOUSE IS now available for the Commodore. It was originally released for the Spectrum and Amstrad computers.

The game will be priced at £9.95 and is obtainable from Melbourne House, Castle Yard House, Castle Yard, Richmond TW10 6TF.



Starion from Melbourne House

Beyond the fringe

BEYOND IS BRANCHING OUT WITH two new games.

The first is the computer version of Superman, a game developed as a joint venture with First Star. Bill Delaney, Beyond's MD said: "It will shoot to number one in every software chart."

Beyond has also formed a partnership with another software house, Nexus. The first game to appear on the new label will be called Nexus and will be a "sophisticated graphic adventure where the player becomes an investigative journalist infiltrating an evil drugs ring in South America.

According to Nexus, the game will offer maximum playability and user friendliness. Instead of having to read pages of documentation before starting, the player will be able to experience action straight away through on screen instructions. Nexus will also feature digitised video images which, it is claimed, will generate realistic animation enabling the player to recognise various characters.

Nexus costs £9.95 and is available from Beyond, Durrant House, 8 Herbal Hill, London EC1R 5EJ.

Gold standards

US Gold has brought out yet another batch of new releases. The three latest titles are Monster Trivia, Beach Head II and Ghostchaser.

Monster Trivia is a spin off from the enormously successful board game, Trivial Pursuits, but in this version you get killed by a monster if you get enough questions wrong.

Beach Head II is the follow-up to Beach Head and features multiscreen play, complex strategy, animation and high speed arcade action.

Ghostchaser takes Harry around Fairport Manor where he must blast large ghosts which materialise at random.

All the new titles are for the C64 on cassette or disk and cost £9.95 and £14.95 respectively. You can get in touch with US Gold at Unit 10, The Parkway Industrial Centre, Heneage St, Birmingham B7 4LY.



Feargal Sharkey sits up and takes notice

Vox pop

POP STAR, FEARGAL SHARKEY, HAS recently become the owner of the first production model of Supersoft's digital sampler for the C64.

He visited the Supersoft stand at the Commodore Computer Show and was very impressed with the sound reproduction quality of the new product.

The sampler is called Microvox and offers eight different sampling rates, up to a maximum of 42KHz which gives a 10KHz band width. Samples can be played forward or backward and there is full editing and looping with a high resolution display.

A 2000 note sequencer is included as part of the Microvox software with real time recording and step time editing.

Microvox costs £299.95 and is available through music shops, computer stores or direct from Supersoft at Winchester House, Canning Rd, Wealdstone, Middlesex HA3 7JS. A disc drive is essential.



Oh joy!

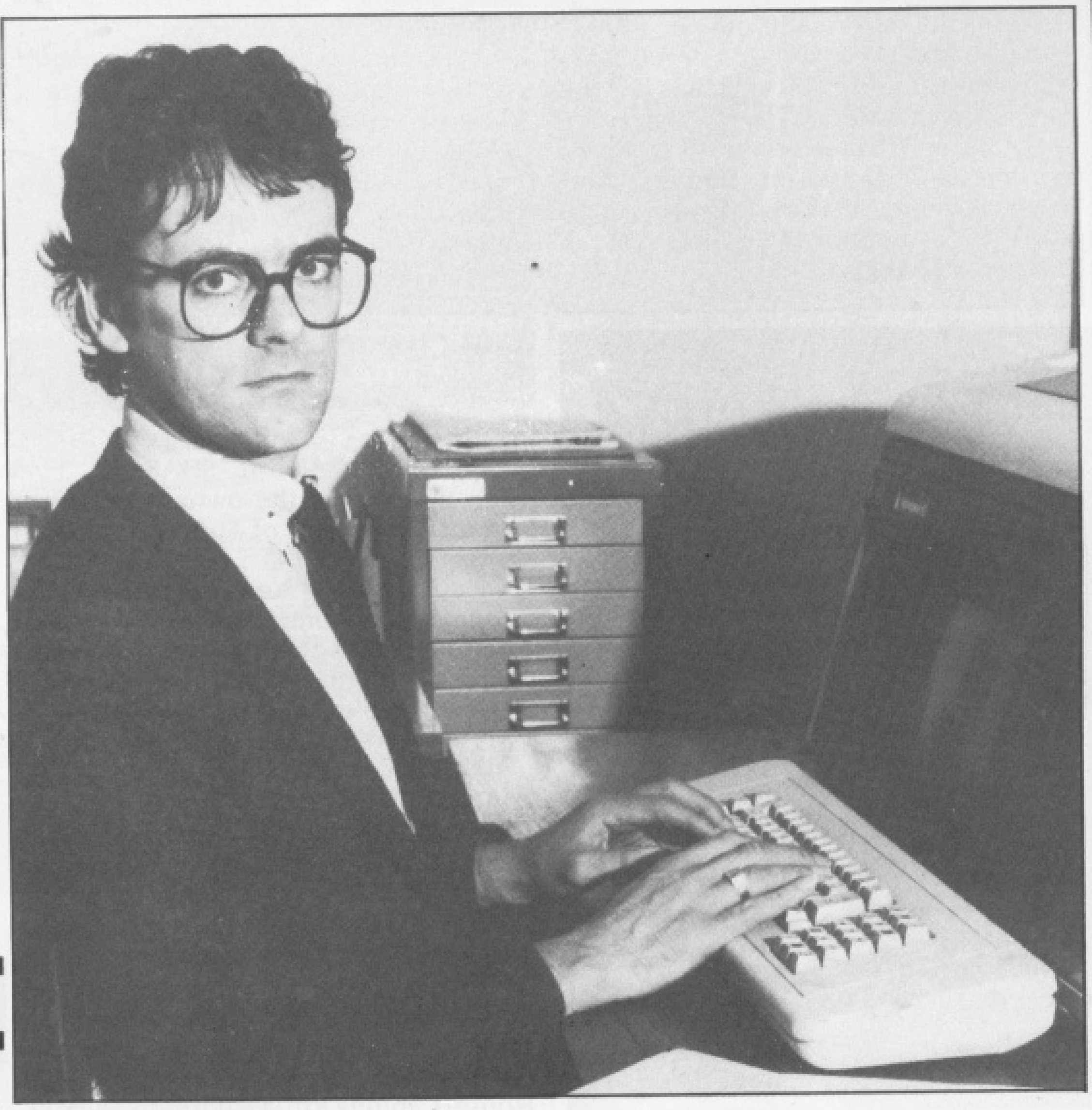
Jumping the gun

C-16 and Plus 4 owners can now pick up a bargain in joysticks with Vulcan Electronics new packages.

The Gunshot I joystick is now being sold complete with an adaptor for £10.95. The packs are available from Dixons, Laskys and other computer retailers.

Everything the user needs to connect the joystick to his computer is included plus concise instructions and a 12 month guarantee slip.

For more information contact: Vulcan Electronics, 200 Brent St, Hendon, London NW4 1BH.



"I'll never have to feel like this again!"

Eye contact

SPECTACLE WEARERS WHO USE VDUs or television screens as a major part of their everyday lives can now try a new way of protecting themselves from eyestrain.

Balzars has come up with Quazar, a tough anti-reflection coating which, claim the makers, virtually eliminates lens surface reflections especially those caused by strong projected images. The coating can increase light transmission to almost 100 per cent.

The result for the user is that extra visual sharpness is noticeable during long periods of exposure, thus reducing visual fatigue.

For more information contact: Balzars, Northbridge Rd, Berkhamsted, Herts HP4 1EN.

Become a boffin

A NEW HOME COMPUTER TEACHING course has recently been released by Fearless Software. It is claimed that the course will take you from scratch to complete computer literally in 12 months.

Peter Ellis who formed the company to produce and market the Home Tutor said: "It is for children, housewives and the retired. It is also for those seeking new jobs or looking for promotion with their present employer."

The course is available on tape or disk and it teaches, demonstrates, corrects and tests the students. There is an examination at the end and those who pass receive a certificate.

The course starts by explaining what a computer is and goes on to teach Basic and machine code, covering such areas as information storage and communications. There are 13 parts to collect over the 12 month period.

Home Tutor costs £12.50 per month including tapes, teaching notes, a carrying case and ring binder. For disks the cost is £2 extra and postage is £1.15. Quarterly charges are £30 for tapes and £35.55 for disks.

For more information contact: Fearless Software, Infoplan, 30 Eastbourne Terrace, London W2.

Errata

A NUMBER OF PEOPLE SEEM TO BE having problems entering the Sketch Pad program which appeared in the September issue of Your Commodore. There are no errors in the program but some of the codes that our printer interface uses seem to be causing the problems.

In line 1450 the [255] is actually the character code for pi (π). This character is to be found on the key next to the RESTORE key. When entering the program you should type in the pi and not [255].

Another line that is causing problems is 620. It appears that in some issues of the magazine this has not reproduced very well. Line 620 should start with IF XC = 256 THEN

A number of people also seem to be having problems finding the ↑ character on their keyboard this is the up arrow ↑ to be found on the key next to the RESTORE key. The printer that we use does not print the downstroke of the arrow.



British Telecom and Program express sign up



The in-store terminal

Telecom deal

BRITISH TELECOM AND PROGRAM EXpress have joined forces to make the lives of software retailers and buyers less fraught with frustration.

Program Express is the firm which launched the Electronic Distribution of Software Machine in July 1984 which works on a "Central Computer Network/In-store Satellite Terminal" principle. The retailer can download a unit of software from the in-store terminal onto a blank tape. The machine records all relevant details about the sale and at the same time new titles can be added to the hard disk terminal while old ones are deleted.

Under the new agreement British Telecom will finance the operation and Program Express's three directors, Gilmour Kennedy, Bruce Neville and Grant Robertson will run the company auutonomously.

INPUT

Agony Aunt Tony Crowther answers more of your programming questions.

INPUT

I have been told on many occasions that it is possible to make all of the keys on the C64 keyboard repeat, as on the Spectrum.

This seems to be a very handy facility as you could enter long strings of the same character without having to press the key for each letter.

However, I do have one slight problem, nowhere can I find mention of how to do this. Would it be possible to provide me with the necessary commands or program to make the C64's keys repeat?

Mike Atkinson
Taunton

OUTPUT

Yes, it is possible to make all of the keys on the C64 repeat and it is very easy to do. All you have to do is POKE into a few memory locations. The locations are as follows

Location	POKE	Result
650	255	all keys repeat
650	0	no keys repeat
650	128	just the cursors repeat
651	0-255	repeat speed
652	0-255	delay before repeat

The keyboard scan is done by the hardware IRQ Interrupt, if we change the clock rate of timer A we can make the rate of the interrupts speed up or slow down. Try:

POKE 56325,number (0 to 255)

a number of zero to 10 forces many interrupts a second and causes the Basic run speed to slow down. This could be used to your advantage when debugging a Basic program as everything will slow down.

INPUT

I write to congratulate you on the quality of your magazine. I find it to be the best of the British Commodore mags available in Australia. However I feel that the atrocious reproduction of graphics symbols spoils it. I would suggest that you employ a system like the American magazine 'Ahoy', which permits rapid loading and clear understanding of listings.

P.Robinson,
Blackwater,
Australia.

I'm not sure what system Ahoy uses but we nevertheless accept that the listings need improving. Therefore we will be using a Micrografix MW350 in future, which replaces all graphic characters with a short description.

INPUT

I have a problem with my 64 that I cannot get an answer to. After about two hours my computer crashes, putting random graphics on the screen and eventually the whole screen is covered in flashing graphics If I turn it off for a couple of hours the problem goes away but it recurs more frequently afterwards.

L.Burn
South Shields
Tyne & Wear

OUTPUT

The problem is that your 64 is overheating I suggest that you take it back to the dealer and see if he can replace it.

INPUT

In your March issue you did a feature on Games creators and this article impressed me so much that I wish now to buy one. The one I am interested in is the Quill by Gillsoft, could you please tell me their address.

P. Vassallo
Pieta
Malta

OUTPUT

Gillsoft are at:-
30 Hawthorn Road,
Barry,
S. Glams.
CF6 8LE.

INPUT

In am writing a program in machine code that requires a lot of data storage. I understand that there is 8K of RAM 'underneath' the Basic Kernal. This would be an ideal place to store my data and machine code routines.

Could you please explain how I access this area of memory as when I try I only get the Basic ROM and not the numbers I have stored beneath it?

OUTPUT

As you have already found out you can't use the area of memory beneath the Basic Kernal when Basic is running as any peeks or calls to that area will just go to the kernal. If however you are using machine code you can turn off the Basic kernal and use the memory that sits beneath it quite easily.

First you must switch off the Basic ROM. To do this you unset the lower bit of memory location one. This can be done simply by subtracting one from it. If this action was to take place in Basic your program would crash. However, in machine code it will not affect the program.

If your machine code routine was to start at — or your data was stored at — location 40960 ($A000), then this small routine would allow you to access it.

```
$C000 DEC $01     ; switch off Basic ROM
$C002 JSR $A000   ; jump to start of
                    program
$C005 INC $01     ; back here from your
                    routine
; first turn Basic
                    back on
$C0006 RTS        ; return to Basic or rest
                    of program
```

OUTPUT

COMMODORE SOFTWARE THROUGH THE POST.

(WITHOUT PAYING THROUGH THE NOSE.)

Simply send off this page and you'll get:

SOFT POST MAGAZINE FREE.



No more traipsing around the shops trying to find the software you want. Let Commodore post it to you. Simply make your selection from our Soft Post magazines which we'll send you every few months (they're packed full of software information) – but there's no obligation to buy anything, if you don't want to.

WIDEST CHOICE OF COMMODORE SOFTWARE ANYWHERE.

In every magazine there'll be special offers: an enormous selection of great software at amazing prices – and not just games, but educational, business and home use, too.

(Sometimes you'll even find peripherals at special prices.)

PLUS ONE OF THESE PROGRAMS FREE.

Tick the box against the software of your choice.

COMMODORE 64
- Robert Carrier's Menu Planner (cass.) ☐
- Know your Own IQ (cass.) ☐
- Patrick Moore's Astronomy (cass.) ☐

VIC 20
- Sargon II Chess (cart.) ☐
- Omega Race (cart.) ☐
- Road Race (cart.) ☐

COMMODORE 16/Plus 4
- Harbour Attack (cass.) ☐
- Space Sweep (cass.) ☐

I would like to receive a regular issue of Soft Post. I own a Commodore (delete as applicable): VIC 20/64/16/Plus 4.

Title (tick): Mr. ☐ Mrs. ☐ Miss ☐ Initials

SO YCO 1110

Surname

Address

Postcode

STD. Code Telephone

OR, IF YOU INTRODUCE A FRIEND.

Fill in your name and address and your friend's below. Now you can choose one item from either list, just tick a box.

VIC 20
- Programmer's Reference Guide (book) ☐
- Introduction to BASIC II (cassette) ☐

COMMODORE 64
- Programmer's Reference Guide (book) ☐
- Fantasy Five (cassette) ☐

COMMODORE 16/Plus 4
- Jack Attack (cartridge) ☐

I would like to introduce my friend who owns a Commodore (delete as applicable): VIC 20/64/16/Plus 4.

Title (tick): Mr. ☐ Mrs. ☐ Miss ☐ Initials

Surname

Address

Postcode

STD. Code Telephone

ONLY ONE APPLICATION PER HOUSEHOLD.
PLEASE ALLOW 28 DAYS FOR DELIVERY.

commodore
HARD SOFT TO BEAT

Post to: SOFT POST, FREEPOST, Commodore Business Machines (U.K) Ltd., Corby, Northamptonshire NN17 1BR.

THE C 128 -IN DEPTH

Stuart Cooke has spent hours slaving over a red hot Commodore 128 and he's decided that it was definitely worth the effort.



COMMODORE ARE ALL SET to prove that three into one does go with their new Commodore 128 computer.

The C128 is essentially three computers in one. Inside a case that looks as though it would be quite at home on the flight deck of some futuristic spaceship there is a good old C64. Secondly we have what can only be described as an upgrade of the C64, 128 mode. Hardware-wise this is the same as the C64 with only a few changes. Thirdly there is a machine that has also been around for quite a long time, but not from Commodore. This is a CP/M computer. CP/M has been around for a number of years and is the name given to an operating system that is used on some Z80 based business systems to allow a great deal of compatibility between manufacturers. Perhaps one of the most famous business packages that runs under CP/M is the wordprocessing package Wordstar which is in use in thousands of offices around the world.

First Impressions

The first thing that you notice about the C128 is that you are not getting some fantastic new machine for your money. What you do get are two very well established machines and one very good up grade.

The machine itself has been designed to achieve as much compatibility as possible with the C64 and other Commodore products. For this reason many of the expansion connectors will be familiar to owners of other Commodore machines. There are connectors on the machine for two joysticks, Commodore's own serial peripheral connector, TV interface, composite video interface, a standard Commodore cassette interface, a user port and a cartridge port. New items obvious on the sides of the machine are a reset switch and a RGB monitor interface.

Even though Commodore has tried to make the C128 as compatible as possible with the C64, the cassette interface is in a silly position. If you plug a Commodore modem into the cartridge port the casing of the modem obscures the cassette interface preventing a cassette recorder from being plugged in.

The keyboard bears only a slight resemblance to earlier Commodore machines as there are more keys available. Added keys include a numeric keypad and a large number of function keys.

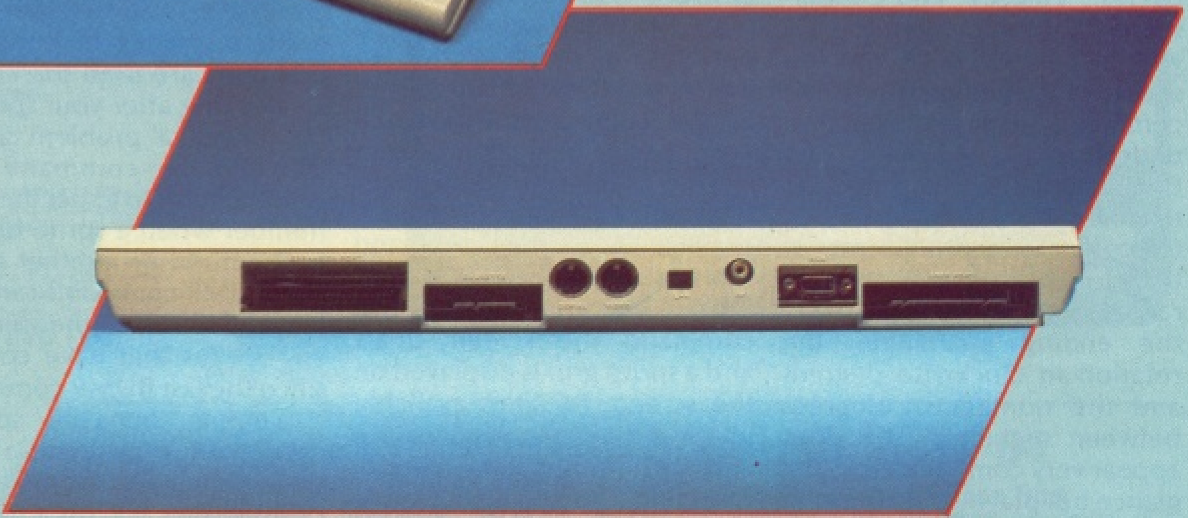
In Use

When powering up you are able to decide whether you wish to boot up the C128 in 40 or 80 columns, CP/M, 128 or 64 modes. Here we come across the first problem with the C128. The 80 column display can only be viewed on a monitor that is plugged into the RGB interface while the 40 column display can only be viewed on a TV or monitor plugged into the composite video socket. If you wish to use both 40 and 80 columns then you will need two different monitors. Commodore has produced a new monitor that will allow you to switch between the two modes of operation. A monochrome composite signal is present on the RGB socket so you can plug a monochrome monitor to view 80 columns if you already own one. Using a monochrome monitor will not cause any problems especially when you realise that the 80 column mode is text only and no graphics are available.

C64 mode offers nothing new, what it does do however is allow you to use all of the packages that are currently available for the C64 on your C128 computer. This is great news for people who already own C64s and are thinking about upgrading. Over 100 pieces of software were tried on the review machine and all of them worked correctly. The only problem that we found was with programs that used keyboard overlays as the keys are in different positions. This is especially noticeable with Commodores Music Maker package as the keyboard will not fit the new design. Even so, it appears that all problems will work with no faults.

An on board Z80 micro processor allows you to run CP/M Plus version 3.0. CP/M Plus will give your CP/M programs full access to the 128K of RAM that is built into the machine as standard. Commodore has made a few changes to CP/M Plus, mostly by adding some new com-





mands. Because of this total, CP/M compatability is something that will have to be proved. Unfortunately the review machine did not come with any CP/M software so this was something that I was unable to test.

As I have already mentioned, 128 mode offers nothing very new apart from an excellent Basic and an 80 column display. Basic 7.0 can only be described as the Basic that should have been implemented on the C64 as it allows total control of all the facilities offered by the hardware without having to resort to POKEs or machine code.

The new disc drive is also a great improvement on the old 1541 when used in C128 or CP/M mode as it will LOAD and SAVE programs around three to four times faster – a great relief to anyone who has used a 1541. Unfortunately, when used in C64 mode the drive works at the same slow speed that we all know and hate.

Basic 7.0

Basic 7.0 contains a large number of commands that are designed to ease the use of sound and graphics together with a large number of 'toolkit' commands. For example the AUTO command will automatically give line numbers. BLOAD and BSAVE allow you to load* and save specific sections of memory. A renumber function helps with the development of long programs. Programs that do not work correctly are easier to debug with the HELP command which will show you where your error has occurred or the TRON and TROFF commands which will display the line number of the line that is about to be executed.

Error trapping is provided by the TRAP command. Whenever an error is detected this command will cause the program to jump to the specified program line rather than 'bombing out'. You will then be able to find out what error has occurred and take the appropriate action. This command will make idiot-proofing your Basic programs very easy.

Finding out what the controllers are doing is also made very easy. JOY will tell you which direction a specified joystick is pointing in and whether the fire button is pressed. The position of the paddles and light pen is also easy to find out with the POT and PEN commands respectively.

Machine code buffs will be pleased to hear that a machine code monitor is included. This will allow you to display, alter and move sections of memory. There is even a machine code assembler and disassembler. The Basic keywords HEX$ and DEC will also prove to be very handy as they will allow you to convert decimal numbers to hex and vice-versa.

A number of new structure commands have been added to the standard FOR/NEXT loop. These include DO/LOOP which will repeat a section of your Basic program UNTIL a specific condition is met or WHILE a specific condition is met. IF/THEN will only allow one statement after the THEN instruction, BEGIN/BEND will allow a number of program lines to be included after a THEN statement.

Graphics

Budding artists are well catered for inside Basic 7.0 by a large number of graphic commands.

GRAPHIC is used to turn on one of the six different graphics modes that are available. The six modes are, 40 column text, standard bit-map graphics, standard bit-map with split screen, multi-colour with split screen and 80 column text. As previously mentioned the 80 column screen can only be used with a monitor plugged into the RGB socket. The split screen modes are very interesting as they allow you to use a section of the screen in one of the graphic modes while still retaining a specified segment for text only. This type of feature has been used many times on the C64, especially in graphic adventures, but is now extremely easy to use on the C128.

Drawing lines or shapes on a graphic screen is also very simple. The COLOR command allows you to set up the colours that you require for any plotting etc. The LOCATE command can be used to position the graphics cursor at any point on the screen, and

DRAW will allow you to plot dots or draw lines in the specified colour. Shapes can easily be drawn as the draw command will allow you to string a number of points together by using the word TO as in

DRAW,100,100 TO 10,100 TO 10,10

BOX makes it extremely easy to draw rectangular shapes on the screen, all that is necessary is to specify the top right hand and bottom left hand co-ordinates together with the rotation of the box. You can even specify if you want the box to be filled in with a certain colour.

CIRCLE is used to draw circles, ellipses and 'other' shapes. This may seem to be a weird statement but when you see how complex the circle command can be you will understand what I mean. The CIRCLE command can have up to nine parameters, these are, the colour source, the centre of the circle, the X radius, the Y radius, the starting arc angle, the ending arc angle, the rotation in clockwise degrees and the number of degrees between segments. This does appear very complicated at first glance but playing around with the parameters will soon show you how versatile this command really is. Not all of the parameters need to be used every time that the command is used. The following example is from the manual and will draw a diamond shape on the screen:

CIRCLE 1,260,40,20,30,,,,90.

WIDTH can be used to set the width of any lines that you are drawing while SCALE will allow you to alter the size of your diagrams with ease. Another handy command is PAINT which will allow you to fill in any area of the screen with a specified colour.

A very limited form of window is implemented on the C128 through the WINDOW command. This allows you to set up a rectangular area on the screen in which all further screen updates will occur. The size of this rectangle can also be set up outside a program by using ESC T to set the top left corner and ESC B for the bottom corner.

As well as having commands for producing pictures on the screen, there are also a few that will tell you exactly what is happening on your display. The RGR will tell you which graphic mode the C128 is in at the moment. RDOT returns the current position of the graphic cursor or the colour of the graphic cursor. You can even find out the window parameters by means of the RWINDOW command.

SPRITES

As with the C64 there are eight sprites available for use on the C128. These can be either hi-res one colour, or multi-colour. However, unlike the C64, you'll never have to perform a single POKE to memory as Basic 7.0 supplies all of the commands that you'll ever need. You don't even need a sprite editor as there is one built into Basic.

SPRDEF is the command that turns on the C128's sprite editor. On entering the command the screen clears and a sprite grid is displayed on the screen. Facilities available in the editor are: turning on and off multicolour mode, changing the colours, expansion of the sprite in X and Y directions, copying sprites and saving sprites. In fact all of the commands that you are likely to need are implemented. One fairly major omission from the sprite editor is the ability to move the sprite around in the grid. You cannot, for example, rotate a sprite or shift it left by one pixel. No doubt someone will develop a routine to perform these commands.

Another way of defining a sprite is to 'draw' the sprite onto the screen using the many drawing commands. The SSHAPE command can then read the sprite data into a string variable. Moving the contents of this string into sprite memory is also made very easy by the SPRSAV command, this moves the specified string into the sprite not required.

The SPRITE command allows you to turn on and off the individual sprites. SPRITE also lets you set the sprite colour, whether it passes over or beneath the background, whether it is hi-res or multi-colour and if it is expanded in either the X or Y directions.

Moving a sprite around is also made child's play with the MOVSPR command. MOVSPR can take a number of forms. It can place a sprite at a specified point on the screen. It can be used to move a sprite to a new position relative to its old one, move a sprite a certain distance at a specified angle and, perhaps its most powerful use, it can set a sprite moving in a specified direction at specified speed and keep it moving. As you can probably see, the MOVSPR command will be a great boon to anyone who wishes to write a game program. Moving your latest deadly creation across the screen can now be done by one command rather than the numerous lines of code that C64 users are used to.

Looking after your sprites is no longer a problem as the COLLISION command will cause a jump to a specified line number when a sprite hits the background or another sprite. The BUMP command can then be used to return the values of any sprites that have collided since the last BUMP command.

Finding out the specific details of any sprite is also made easy with a number of commands. RSPCOLOUR returns the colour of a specified sprite. RSPOS will return the X, Y co-ordinates of a sprite, very handy after a BUMP instruction, and RSPRITE will tell you whether a sprite is on, off, expanded etc.

As you can no doubt see from the brief mention of the sprite manipulation commands that are mentioned above, writing any sort of program that uses sprites is now extremely simple. Before very long we should start to see some excellent graphic programs written totally in Basic.

Sound

The sound chip that is used in 128 mode is exactly the same as in the C64, the major difference being that there are now a large number of commands available to make control of it easy. There are five commands available for sound manipulation. SOUND is used to place quick and easy sound effects in your programs. SOUND can have up to eight parameters. These are, the voice number (one to three), the frequency of the note, the duration of the note, whether the sound is to be incremented or decremented while playing, the minimum frequency the note can go to, how big the step up or down is, the type of waveform to use and the pulse width if you are using a square wave. As you can see, some very interesting effects can be made by using this command. Another command is designed to make the playing of music easy, this is the PLAY statement. PLAY allows you to set up a string of synthesiser control characters inside quotes. Characters allowed are the musical notes ABCDEFG, characters which tell the synthesiser what type of note is playing and characters to specify the voice, octave, envelope volume and filter. Ten predefined envelopes are available ranging from harpsichord to xylophone and you are able to define your own using the ENVELOPE command. TEMPO defines the speed of the song being played and VOL the volume. The command FILTER will also let you set up the filters parameters very easily.

Adding a musical accompaniment and sound effects to your programs is now very easy. Even a beginner will soon be producing sounds that are the same quality as those that the top programmers have been producing on the C64.

Verdict

Obviously I have not been able to cover all of the details of the C128 and its Basic, there are many commands that I have not covered. However, from the few that I have mentioned I think it should be fairly obvious that the C128 is a powerful machine, or should I say machines? The fact that it will run all C64 software, and that thousands of business packages are available with CP/M, make the machine a bargain for the beginner, hobbyist and businessman alike.

Commodore seems to have a winner.

U.K. Soft Centre

COMMODORE SOFTWARE SPECIALISTS

C64 Business		RRP	2U	
Calc Result (Adv) by Handic		99.95	**85.00**	c
Calc Result (Easy) by Handic		49.95	**39.75**	c
DFM Database by Dialog		24.00	**20.45**	d
Practicalc II		69.95	**59.95**	d
Superbase 64 by Precision		99.95	**85.00**	d
Superbase Starter		39.95	**35.95**	d
Superscript by Precision		69.95	**59.95**	d
Triangle (Int. DB.SS. WP)		19.95	**17.50**	d
Viazastar 64 XL4		99.95	**89.00**	c
Forecaster*/Project Planner*/Entrepreneur				
Decision Maker* by Brainpower		19.95	**17.50**	t

C64 Educational			C64	
Mr T Series	SPECIAL	9.95	**8.50**	t
Commodore O Level S'jects		9.95	**8.75**	t
French Mistress A & B		17.90	**14.50**	t
German (Linkword)*		12.95	**10.95**	t
Music Studio*	NEW	14.95	**12.75**	t
Get Ready to Read (3yrs+4 tapes)		11.95	**10.45**	t

C64 Flight Simulators/War Games			C64	
Battle for Britain	NEW	9.95	**8.50**	t
F15 Strike Eagle*		14.95	**12.75**	t
Fighter Pilot*		9.95	**8.50**	t
Heathrow Air Traffic C'trol		7.95	**6.50**	t
Kennedy Approach*	NEW	14.95	**12.75**	t
MIG Alley Ace*		9.95	**8.50**	t
Skyfox*	NEW	9.95	**8.50**	d
Sublogic Flight Simulator II		49.95	**44.95**	d
Sublogic Flight Simulator II		39.95	**35.95**	t
Theatre Europe		9.95	**8.50**	t
747 Flight Simulator (Docsoft)		12.95	**10.45**	t

C64 Utilities/General			C64	
Diary 64 by Handic		19.95	**17.95**	c
Doodle		14.95	**12.75**	d
Jetpack Compiler*		14.95	**11.95**	t
LOGO	SPECIAL	34.95	**19.95**	d
Machine Code Tutor		14.95	**12.75**	t
Simon's Basic		49.95	**41.90**	c
Supertype (Typing course)		19.95	**17.50**	t
The Quill		14.95	**11.85**	t
White Lightning*		19.95	**16.50**	t

C64 Hardware			C64	
Quickshot 2 (Joystick)		11.95	**8.75**	
Supersketch* by Anirog		49.95	**47.50**	t
Voicemaster* by Anirog		59.95	**57.40**	t

C64 Adventures		RRP	2U	
Borzak	NEW	6.95	**5.95**	t
Dragonworld		19.95	**18.00**	d
Fahrenheit 451	NEW	19.95	**18.00**	d
Fourth Protocol	NEW	12.95	**10.95**	t
Hobbit*		14.95	**11.85**	t
Macbeth		14.95	**11.85**	t
Murder on the W/front*	NEW	14.95	**12.75**	t
Red Moon	NEW	6.95	**5.95**	t
Seven Cities of Gold	NEW	14.95	**12.95**	d
Swiss Family Robinson	NEW	14.95	**12.75**	d
Tracer Sanction	NEW	19.95	**17.95**	d
Treasure Island	NEW	14.95	**12.75**	d
Ultima III	NEW	19.95	**18.00**	d

C64 Action/Games/Sports			C64	
Archon*		11.95	**9.95**	t
A View to a Kill		10.95	**8.95**	t
Beachhead II	NEW	9.95	**8.50**	t
Blackwyche	NEW	9.95	**8.50**	t
ELITE*		14.95	**12.75**	t
G. Gooch's Test Cricket		9.95	**8.50**	t
Gribbly's Day Out		7.95	**6.50**	t
Mercenary*	NEW	9.95	**8.50**	t
Monty on the Run	NEW	7.95	**6.50**	t
My Chess II*		11.95	**10.50**	t
Nick Faldo's Golf		9.95	**8.50**	t
Nodes of Yesod	NEW	9.95	**8.50**	t
Racing Destruction Set	NEW	14.95	**12.95**	d
Rescue from Fractulus*	NEW	9.95	**8.50**	t
Rupert & Toymakers Party	NEW	7.95	**6.50**	t
Scrabble*		12.95	**11.00**	t
Speed King*	NEW	9.95	**8.50**	t
Stealth	NEW	9.95	**8.50**	t
Summer Games II*	NEW	9.95	**8.50**	t
Thing on a Spring		7.95	**6.50**	t
Wizardry	NEW	9.95	**8.50**	t
World Cup II	NEW	7.956.50 t		

C16 General			C16	
Airwolf	NEW	6.95	**5.95**	t
Berks I, II and III	Special	20.85	**16.00**	t
D. Thompsons Star Events	NEW	7.95	**6.95**	t
Defence 16		7.95	**6.95**	t
Psychedelia		7.95	**6.50**	t
Spiderman		7.95	**6.50**	t
The Boss	NEW	6.95	**5.95**	t
Timeslip	NEW	6.95	**5.95**	t
World Cup	NEW	6.95	**5.95**	t

t = cassette tape d = diskette c = ROM cartridge *= Disc version available

HUNDREDS MORE TITLES AVAILABLE — **ALL** AT DISCOUNT PRICES

MOST **C16 PLUS 4** and **C128** titles available — ALL with similar discounts to C64

Please send stamp for our complete list. (state which computer).

Please include 50p for Post and Packing for all orders less than £20. Outside UK please include £1 plus 75p for each additional item on all orders. Send Cheque, Eurocheque, Sterling money order of UK Postal Order with order to:

PO Box 36
DUNSTABLE
Beds LU6 2NP

UK SOFT CENTRE LTD (YC)

Telephone:
DUNSTABLE
(0582) 607929

SJB DISKS LIMITED

BLANK DISKS

Top Quality 5¼" Bulk Packed Disks

Supplied in 10's with a **FREE** Plastic Library Case, Labels etc.

10 Single Sided/Double Density*	ONLY **£13·95**
10 Double Sided/Double Density*	ONLY **£17·95**
50 Single Sided/Double Density* (Supplied with a Perspex Storage Box)	ONLY **£59·95**
50 Double Sided/Double Density* (Supplied with a Perspex Storage Box)	ONLY **£74·95**

★ LIFETIME WARRANTY ★
★ FREE FAST DELIVERY ★

All prices are inclusive of V.A.T.
Delivery - **FREE** throughout the U.K.

3" & 3½" Disks also available

Bulk Orders, Export Enquiries & Educational Enquiries Welcome

Double Density is the equivalent of 40/80 Track

Please Send Cheques/Postal Orders to:-
SJB DISKS LIMITED **(Dept Y.C.)**
11 Oundle Drive, Nottingham, NG8 1BN
Telephone (0602) 782310

cascade CASSETTE 50



50 COMPUTER GAMES ON ONE CASSETTE

YOURS FOR £9.95

including FREE Watch and Postage

CASSETTE 50 IS AVAILABLE ON

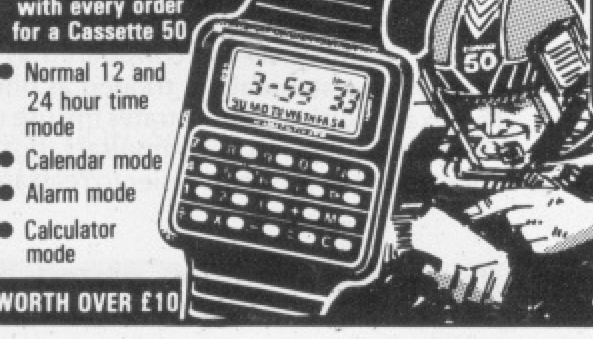
BBC A/B ATARI DRAGON ELECTRON VIC-20 ORIC-1 ZX81 apple Atmos Spectrum commodore 64 AMSTRAD

FREE

20 KEY, MULTIFUNCTION CALCULATOR WATCH with every order for a Cassette 50

- Normal 12 and 24 hour time mode
- Calendar mode
- Alarm mode
- Calculator mode

WORTH OVER £10



FREE *calculator watch with every Cassette 50*

EXPRESS DELIVERY-ORDER NOW!

ONLY £9.95

Commodore 64

Only you can save Europe from destruction! It's ROCKET LAUNCH, the thrilling war game that reproduces a European map. More ambitious? Try rescuing your crew under an ultrafast GALACTIC ATTACK and escaping back to your spaceship! Just two of the great games on your Cassette-50, featuring high resolution and user-defined graphics, sprites, sound and music.

1. Maze Eater
2. Galactic Attack
3. Space Mission
4. Lunar Landing
5. Plasma Bolt
6. Startrek
7. Radar Landing
8. Attacker
9. Galactic Dog Fight
10. Psion Attack
11. Ivasive Action
12. Noughts & Crosses
13. Boggles
14. Pontoon
15. Ski Jump
16. Hangman
17. Old Bones
18. Thin Ice
19. Orbitter
20. Motorway
21. Force Field
22. Nim
23. Tunnel Escape
24. Barrel Jump
25. Cannonball Battle
26. Overtake
27. Sitting Target
28. Smash the Window
29. Space Ship
30. Jet Flight
31. Phaser
32. Intruder
33. Inferno
34. Ghosts
35. Submarines
36. Rocket Launch
37. Planets
38. Black Hole
39. Dynamite
40. Do Your Sums
41. Derby Dash
42. Space Search
43. Universe
44. Rats
45. Tanker
46. Parachute
47. Jet Mobile
48. High Rise
49. The Force
50. Exchange

Names and games may vary for each type of computer.

Postage FREE in U.K. Add £1.00 for overseas orders. Free watch and tape will be despatched within 7 days.

I enclose a cheque/postal order for £ ______ made payable to Cascade Games Ltd.

or through any Post Office by TRANSCASH (Giro No. 655 6655)
For even faster ordering when charging to Access, Barclaycard and Trustcard Visa use our 24 hour service. (0423) 504663.

No. ______

COMMODORE 64 ☐	ATARI ☐	ORIC 1 ☐	BBC A/B ☐
SPECTRUM ☐	VIC 20 ☐	ZX 81 ☐	APPLE ☐
AMSTRAD ☐	ATMOS ☐	DRAGON ☐	ELECTRON ☐

Name ______
Address ______
Post Code ______
Country ______ YC11/85

Cascade Games Ltd., 1-3 Haywra Crescent, Harrogate, North Yorkshire, HG1 5BG, England.
Telephone: (0423) 504663. *Registered Number 1755554*

Our resident linguist, David Janda, gives you a breakdown of Pascal packages for the C64.

Language Lab

P•A•S•C•A•L

PASCAL WAS INVENTED BY ONE MAN, Niklaus Wirth of the ETH Technical Institute of Zurich in 1970. It is a compiled language that was designed as an aid to teaching good programming practice.

Because the language is very concise, institutions found it easy to implement on their systems. Software houses also discovered that it was possible to implement Pascal on many home micros, hence the reason for its early appearance on the micro scene.

Program Body

Unlike Basic, where you have a free hand in program structure, Pascal requires the programmer to 'section' his programs. There are three main sections to a Pascal program.

```
PROGRAM – header declarations
BEGIN – Main body
END
```

The first section is the program header. Every Pascal program must start with the reserved word 'PROGRAM', which is followed by the name of the program. This can be optionally followed by I/O declarations, implemented in various ways in different compilers.

The next section is the declarations. There are a number of these. First there is the reserved word 'CONST', used to define a symbolic constant:

```
CONST
  PI=.14159;
  AGE=23;
```

In the example, two constants have been declared. Following the constant declarations, come the more common variable declarations:

```
VAR
  PI   :   REAL;
  SUM,WEIGHT,TOTAL : INTEGER;
  SWICH : BOOLEAN ;
  INITAL : CHAR;
```

The four data types are integer, real, string and boolean. Pascal requires all variables to be declared explicitly. For Basic programmers this may come as a bit of a shock. It's very easy in Basic to declare yet another variable as you need it, but in Pascal this is not the case. The good point in declaring variables is that you need to do some thinking and plan on how many variables you will need in the first place!

The next declaration is probably one of the most powerful features of Pascal – type definition. As you can see from above; there are four data types in Pascal. These are pre-defined data types, and if you wish you can declare more:

```
TYPE
  DAY =(SUNDAY, MONDAY, TUESDAY,
WEDNESDAY, THURSDAY, FRIDAY,
SATURDAY);
```

To make this a little clearer imagine the following:

```
TYPE
  INTEGER=(1,2,3,4,........65534,65535);
```

In other words, defining your own data type is a convenient means of giving a name to an ordered sequence of known elements. To add the icing to the cake, a SET can be declared which is a collection of objects of the same type:

```
TYPE
  CAPS=SET OF 'A..Z';
```

Now we can have the following in a VAR list:

```
VAR
  LETTER : CAPS;
```

The last two items in the declarations section of a Pascal program are procedures and functions. Basically, a procedure is a sub-program (a sort of subroutine), which contains other statements and so on. Functions are a means of declaring new operations that Pascal does not have.

So far, the main program body has not been discussed. As you might have already guessed, Pascal programming requires a lot of thought. This is a good thing because people tend to program at the keyboard without thinking first. Thinking about what data types and variables you'll need will encourage thought, and a good program can be the final result.

Pascal Syntax

The third section in a Pascal program is the main body of the program. Here, the first and last reserved words are BEGIN and END, with the final END followed by a full stop. I say 'final' because there are normally more than one set. To understand this better, here is a very simple Pascal program:

```
PROGRAM Greeting;
BEGIN
  WRITELN('Hello Your Commodore
readers!')
END
```

Notice that there is no declaration section. The two statements in our small example are the first line and WRITELN. Statements are separated by a semi-colon. The exception to this case is the statement preceding and END:

```
PROGRAM Greeting;
BEGIN
  WRITELN('Hello');
  WRITELN('How are you today?')
END
```

In the example, the first statement after BEGIN ends with a semi-colon because the next line contains another statement. Now look at this example:

```
PROGRAM Count;
VAR
  I : INTEGER;
BEGIN
  WRITELN ('Watch this!');
  FOR I:=1 TO 10 DO
    WRITELN('Ying')
END
```

This simply prints 'Watch this' followed by 'Ying' printed 10 times. However, if I wanted to print 'Ying' followed by 'Yang' on the next line a nested BEGIN..END has to be used:

```
PROGRAM YingYang;
VAR
  I : INTEGER;
BEGIN
  WRITELN('Watch this!');
  FOR I: =1 TO 10 DO
    BEGIN
      WRITELN('Ying');
      WRITELN('Yang')
    END;
WRITELN('That's all folks.')
END
```

The two statements between the second BEGIN..END are considered as a compound statement i.e. treated as one. This example also demonstrates the use of identation. This is NOT compulsary, but it does make the program easier to read and

follow through logically. The normal rule of thumb is that when nesting occurs (more BEGINs) you indent, and as the END is matched on the same column indentation occurs:

```
BEGIN
 statements
  BEGIN
   statements
    BEGIN
     statements
    END
  END
END
```

Procedures and Functions

Tables one and two list the standard Pascal reserved words as well as pre-defined procedures and functions. There are not many, but they are pretty powerful. Pascal provides the features to define your own procedures and functions made up from existing procedures and functions.

In essence, procedures and functions are mini-programs. They may have their own declarations and program blocks just like the main program. Although similar, there are a couple of differences between the two. Procedures do not have to have a parameter passed to them, although it is possible to pass parameters to and from procedures. Functions on the other hand, must have a parameter and can be used in comparisons whilst procedures cannot.

Here is an example of a program that uses a procedure. All it does is print the numbers one to 10, 10 times:

```
PROGRAM Test;
VAR
 I : INTEGER;
 PROCEDURE Count;
 VAR
  I : INTEGER;
 BEGIN
  FOR I:=1 TO 10 DO
   WRITE(I)
 END;
BEGIN
 FOR I:=1 TO 10 DO
  Count
END
```

Notice that the procedure and the main program use a variable with the same name. This is OK, because variables are local to procedures and functions. Also note that the procedure 'Count' is called from the main program just like any other procedure (no line numbers like Basic!).

A function is declared in a similar manner, although its operation is different:

```
PROGRAM Table;
VAR
 I : INTEGER;
 FUNCTION Square (N : INTEGER) :
 INTEGER;
 BEGIN
  N:=N*N
 END;
BEGIN
 FOR I:=1 TO 10 DO
  BEGIN
   WRITE(I);
   WRITELN(Square(I))
  END;
END
```

In the example, I have used the function 'Square' directly, but as with normal functions it is possible to pass the result to a variable.

Control Structures

Commodore Basic is a bit limited as far as control structures are concerned. Pascal on the other hand, offers the programmer some very elegant means of controlling program flow.

One control structure which has already been covered in the examples is the FOR..DO loop, which is similar to the Basic FOR..NEXT:

```
FOR i:=1 TO 10 DO
 WRITELN(I);
```

DOWNTO is used to reverse the loop:

```
FOR I:=10 TO 1 DO
 WRITELN(I);
```

To include more than one statement within the loop boundaries, a compound statement is used:

```
FOR I:=1 TO 20 DO
 BEGIN
  Statement;
  Statement;
     "
     "
     "
     "
  Statement
END;
```

One last point about a FOR..DO llop is that unlike Basic, Pascal does not like the index (the variable after the FOR) to be altered within the loop itself.

Probably the major drawback with the FOR..DO type of loop is that you have to specify an end to the loop. That is, a FOR..DO loop must have fixed boundaries – even if passed by variables. Pascal offers a couple more control structures which are more flexible.

The first of these flexible control structures is REPEAT..UNTIL:

```
REPEAT
 A:=A+1;
 WRITELN(A)
UNTIL A=10
```

Notice that A would have been initialised before entering the REPEAT loop. Also note that compound statements do not have to be used. Instead, statements are merely separated by the semi-colon. Can you see why this is the case?

You can see from the example that a boolean test is performed after the UNTIL. Any of the boolean operators can be used in this test including ' ', '=' and so on. The most important point worth remembering about the REPEAT..UNTIL loop is that the statement(s) within it will be performed at least once. This is because the test is done at the end of the loop.

The second 'ever-so-flexible' control structure is the WHILE..DO loop which takes the following form:

```
WHILE level desirable DO
 BEGIN
  level:=level+1;
  mix
 END;
```

Notice that the boolean test is performed before any statements are executed so that if the boolean test is false no statement will be executed. A final point to note is that, unlike the REPEAT..UNTIL loop, multiple statements must be treated as compound statements i.e. with a BEGIN..END.

The final control structure to be considered is the CASE statement. This is used in situations where the number of alternatives is greater than two. It is best understood by example;

```
CASE month OF
   1:WRITELN('January');
   2:WRITELN('February');
   3:WRITELN('March');
  12:WRITELN('December')
END;
```

In the example the variable 'month' has a certain value. Depending on what that value is, perform a different action. That is exactly what the CASE statement does. If 'month' is equal to five, then 'May' will be printed, and so on.

Oxford Pascal
Oxford Computer Systems
Disk

Not all compilers produce native machine code, some produce what is known as P-code. The version of Pascal from Oxford Computer Systems does just that. The result is an executable program that doesn't run very fast. However, it should be noted that there are numerous Pascal P-code compilers because they are easy to implement.

Although Oxford Pascal is quite expensive, you do get a lot for your money. First, it follows the Pascal standard almost to the letter. Because of this it

could be used as a serious Pascal training tool.

A major problem with disk based compilers is the time it takes from writing the program to running it! Because of this, Oxford Pascal offers two methods of running Pascal programs.

The first method is to use the resident compiler. This is the default option when the compiler is first loaded. In this mode it is possible to write, amend, compile and run a Pascal program without having to access the disk drive. This allows learners to get their feet wet and generally muck about.

The second method of operation is disk mode. Here, a program has to be developed in the standard method. That is, entered, saved to disk, compiled and re-edited if any errors occur. This mode does offer numerous advantages though. Programs can be much larger as compilation is from disk. Other advantages include data file handling, the availability of the full compiler syntax, external procedures and so on. Object code can also be converted to run as stand alone code.

Machine dependant features include a form of peek and poke, colour and sound and some graphic commands. Most notable of these is the window command that allows the screen to be split between the high-resolution graphic screen and the text screen.

Pascal 64
Orpheus Ltd
Tape

Pascal 64 and Oxford Pascal are so similar in many ways that most of the comments in these reviews apply to both packages.

As with Oxford Pascal, Pascal 64 offers an almost complete definition of Pascal as defined in the User Manual and Report by Wirth & Jensen. It is however, a cassette version and does not produce stand alone code (i.e. does not have a disk mode).

The Pascal source is entered in an editor which is very similar to the standard C64 editor.

There are a few differences and additions. Namely, source code can be entered with indentation, thus showing the program structure. Additions to the standard editor include commands such as search and replace, auto line numbering, renumber, delete and so on.

The source program can be compiled with or without a listing, which can be directed to the printer. The object code can then be run and/or saved to tape. But as stand alone programs cannot be produced, it is necessary to have the compiler in memory when running any object code.

Additions to the standard are similar to what Oxford Pascal offers. But the general purpose DRAW command in Pascal 64 is much faster (see benchtests). Additional features include sprite handling, the ability to read a light-pen and the joystick ports.

Both Oxford Pascal and Pascal 64 have common faults. First, both packages are slow (see benchtests). In fact neither of the two compilers is that much faster than Basic. This is a great pity as one of the advantages of Pascal is its speed.

Other problems include flickering of the high-resolution graphics screen. But the biggest fault is that this screen is divided into 250 by 200 pixels, why not the full 320?

Pascal 64
First Software
Disk

When I first heard about this package I though; "Great, a true Pascal compiler." However, after using it I have mixed feelings; it's both very good and really awful.

The package consists of a disk and 70m page poorly written manual/user guide. The software was written by a German company called Data Becker and the text of the manual and software have suffered through translation!

The compiler is very disk intensive and takes about three stages before a program can be run. The tasks involved in developing a program with this package are; write the source, compile the source, link the object code and save object code to disk. On the good side the compiler produces compact stand-alone code, but don't expect to achieve that in five minutes.

The biggest let-down with this package is the first stage of producing a program. There is no editor and it is necessary to enter the source code using the standard editor. Not only is this a bother (there are no extra editing commands), but it is a disaster. The reason for this is simple. Part of the 'beauty' of Pascal is the ability to write programs that are indented. This makes the program look neater, easier to read and identify. Because Pascal 64 requires the source to be entered using the standard '64 editor no indentation is allowed (unless spaces are preceding by a ';' which makes the source look ugly!).

Another weird thing about the package is that all Pascal keywords must be followed by a space character. This is non standard, and the more experienced programmer will be frustrated by this, whilst the beginner will pick up a bad habit. Why a space is required, is beyond me.

Although the review so far has been negative, things get better after the source has been entered. Once the code has been compiled and linked a machine code file is produced and can be run stand-alone.

The compiler itself is the best of the three packages reviewed, offering more advanced features. These include comprehensive file handling, external procedure support, high-resolution plot, sprite command etc.

An integer command enables faster execution if only integers are being handled. Another impressive feature is the ability to run a procedure as an interrupt. The constraints on this are rather limited, but it could be useful.

Table 1 — Pascal Reserved Words

Here is a list of Pascal reserved words. Note that this list is the required set of words and that some versions may have more - but not less.

AND, ARRAY, BEGIN, CASE, CONST, DIV, DO, DOWNTO, ELSE, END, FILE, FOR, FUNCTION, GOTO, IF, IN, LABEL, MOD, NIL, NOT, OF, OR, PACKED, PROCEDURE, PROGRAM, RECORD, REPEAT, SET, THEN, TO, TYPE, UNTIL, VAR, WHILE, WITH

Table 2 — Standard Functions and Procedures

Every version of Pascal should have the following functions and procedures (with the possible exception of NEW and DISPOSE). All implementations of Pascal on the 64 have more pre-defined procedures which deal with colour and so on. Note that the brackets indicate an argument.

GET(), PAGE(), PUT(), READ, READLN, RESET(), REWRITE(), WRITE, WRITELN, ABS(), ARCTAN(), COS(), ESCP(), LN(), SIN(), SQR(), SQRT(), EOF(), EOLN(), ODD(), CHR(), ORD(), ROUND(), TRUNC(), PACK(), UNPACK(), NEW(), DISPOSE(), PRED(), SUCC()

Summary

If you intend to learn computer science then I would strongly recommend that you get a Pascal compiler for your 64. Pascal has a small amount of reserved words, yet many of the up and coming fifth generation languages incorporate Pascal-like structures.

As far as recommending one of the three packages I can only suggest this. If you wish to learn Pascal, then Oxford Pascal is by far the best. Even though 'Orpheus' Pascal '64 is very similar, the disk operation within Oxford Pascal is a must. However, Oxford Pascal is a bit pricey, so Orpheus' Pascal 64 would be a very good second.

For the more experienced programmer I would recommend First Publishing's Pascal 64. Even though the user interface and daft syntax requirements are off-putting. I say this because the results are quite fast, and it does have a number of advanced features not found in the other packages.

David Janda is prepared to discuss this subject further through electronic mail. He can be contacted on the following

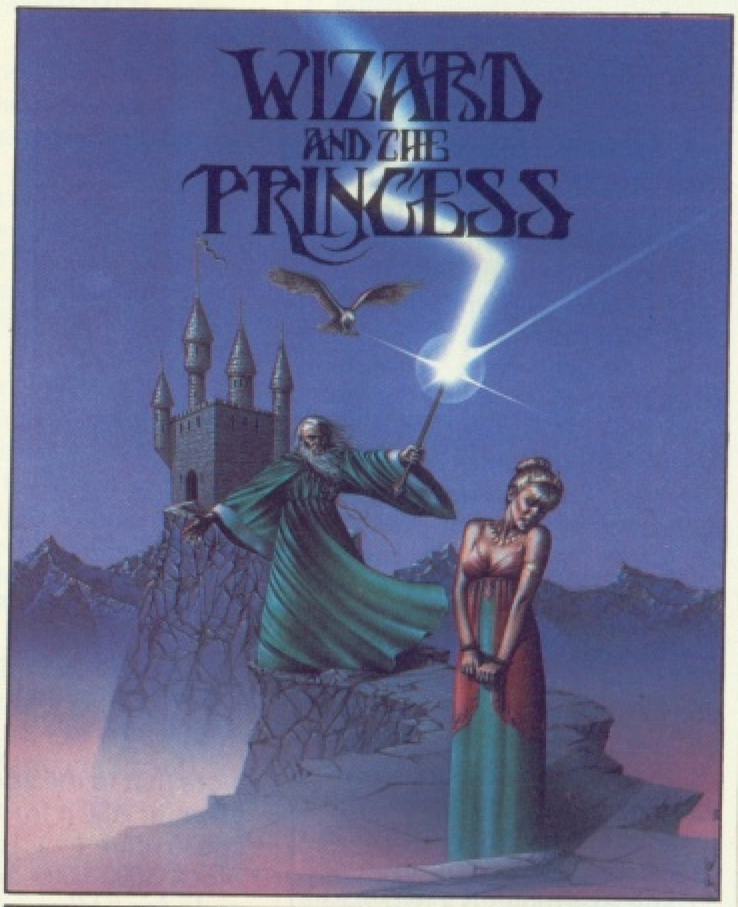
One-To-One: 13419001
Prestel: 919992677

BORED WITH SHOOT 'EM UP SPACE INVADERS?

then it's time to

64 VENTURE INTO THE UNKNOWN...

Enter the world of computer adventure. Each adventure takes several weeks to master, providing hours of fascination and intrigue. You must use your imagination and logic to unlock the mysteries of each adventure. Map skills are essential since each Sierra adventure contains over a hundred full-colour computer generated images which lead you in every direction. You'll need to muster up all your wit and logic and perseverance for each Sierra adventure will prove a test to your imagination and ingenuity. . .



THE WIZARD AND THE PRINCESS

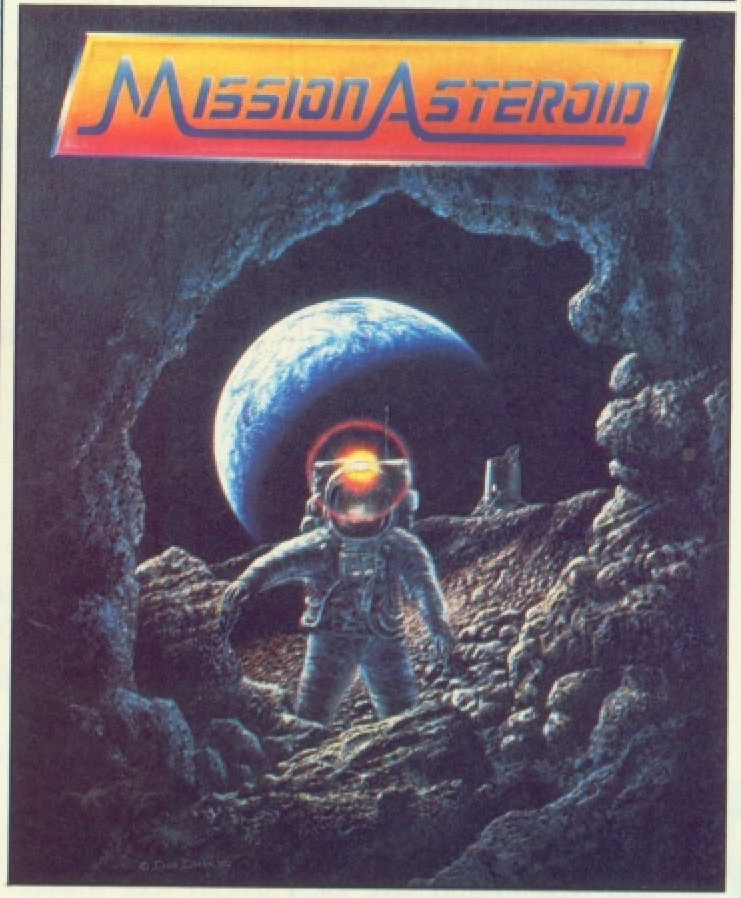
Become the hearty and steadfast adventurer who rescues the fair princess from the dreadful wizard, Harlin. He has taken her to his towering castle beyond the great mountains. You must outwit and outmanoeuvre the mystical power of the wizard which has grown dark and potent through time. Begin your adventure in the village of Serinia, but take heed, everything is not as it may appear. . .

ULYSSES AND THE GOLDEN FLEECE

Protected by the gods, the 'Golden Fleece', legendary treasure of ancient mythology, has been kept from man for many decades. Many valiant men have attempted its recovery, but none have been successful in possessing it for any length of time.

As Ulysses you are assigned to the task of retrieving the 'Golden Fleece' and return it to the King. You set sail from Ancient Greece and on your voyages encounter many of the foes that troubled your predecessors; the wicked Sirens, the one eyed Cyclops; Pluto, mythical god of the underworld and other deadly enemies. Legend acclaims Ulysses as the most daring and skilled seaman of all time. Clad in a pocketed Toga and chain armour, it's up to you to prove it!





MISSION ASTEROID

A terrible tragedy is about to happen...An asteroid of incredible size is headed for a direct collision with the earth. The predicted impact could destroy all life—unless something is done quickly.

It's your duty to fly to the asteroid in a brave attempt to alter its course. But time is of the essence! You must travel quickly and carry out your tasks precisely if disaster is to be avoided. You have a predetermined time in which to achieve your objective. Be quick and careful . . . the future of the earth is in your hands alone!

Sierra ON-LINE INC

All American ADVENTURES

All American Adventures Limited,
Unit 10, Parkway Industrial Centre,
Heneage Street,
Birmingham B7 4LY.
Telephone: 021-359 8881.
Telex: 337268.

DISK ONLY

£14.95

MACH 1

This month we begin a new series by Steve Carrie. When you've collected all three parts you'll have a full machine code development system.

THE AIM OF THIS SERIES OF programs is to provide the reader with a set of tools to aid in the production of 6502 machine code programs on a C64 suitably equipped with a1541 disk drive.

There are three separate programs; a Machine Language Monitor/Editor, a Macro Processor and an Assembler. The Monitor is essentially the control program from which the other two may be called upon to process source code written using the Editor.

The Macro Processor is not a full implementation since it does not handle conditional processing. However, it does allow sections of source code to be written as a "macro" with parameters for later inclusion in source code programs.

The Assembler is a two-pass type which produces an executable disk file which may be loaded into memory by using a simple Basic command:

LOAD "filename",8,1

The programs are disk-based since it was necessary to have the random access and reliability that a disk drive affords.

I will first deal with the Monitor/Editor program and then the Macro Processor and Assembler in turn.

The Monitor

The four Basic listings, when loaded and RUN in sequence 1-4 will POKE the complete Monitor/Editor program into memory. Each listing contains REM statements to show where the various sections of code begin and end. I suggest that you type in and save all four programs before you attempt to generate the code. A word of caution here; check your typing carefully as no checksum facility is provided.

When all four are ready to be RUN, load and execute them each in turn then give the following Basic direct command:

POKE43,0:POKE44,130:POKE 45,128:POKE46,142:SAVE" MONITOR",8,1

Having done this, type:

POKE43,1:POKE44,8:NEW

You should now have a copy of the actual program on disk and are ready to test it out.

Monitor Command Summary

Command	Description
+pc,sr,ac,xr,yr,sp	input processor register values. The R command outputs a similar format allowing free use of the C64 screen editor.
*addr,byti,...byt8	input 8 consecutive byte values into memory starting at addr. The M command outputs a similar format allowing free use of the C64 screen editor.
I	initialises, performs a warm start.
M addr1,addr2	display contents of memory between addr1 and addr2.
R	displays processor register values.
X	exits to Basic command mode.
G addr	restores processor register values and begins execution at addr.
T addr1,addr2,addr3	copy the contents of memory block addr – addr to a block startling at addr3.
F addr1,addr2,byt	fill the memory block addr1-addr2 with the value byt.

Note that in the above commands, the value byte may be given in ASCII format by preceding the character by the apostrophe (   ). For example, F 5000,6000, A will fill the block 5000-6000 hex with the value 41 hex, the ASCII code for "A".

L filename	loads a file into memory at its original address.
S filename addr1,addr2	save the memory block addr1-addr2 to a disk file with name filename. Filenames should not contain spaces.

Monitor Command Summary — Multiple Letter Group

RESET	causes the machine to cold start
DIR	displays current disk directory
CAT	displays current disk directory
DERR	displays disk drive status
INIT	initialises drive (disk I command)
COMPACT	validates disk (disk V command)
FORMAT diskname,ID	formats disk (disk N command)
COPY filename=filename	copies a file (disk C command)
RENAME filename=filename	renames a file (disk R command)
SCRATCH filename	deletes a file (disk S command)
HEX	sets default I/O mode to hex
DEC	sets default I/O mode to decimal
TTY	defaults output device = PRINTER
VDU	defaults output device = SCREEN
EDITOR	enters Editor mode

Testing the Monitor

Enter the monitor by issuing the direct Basic command:

SYS 33280

You should be given the message:

COMMODORE 64 UTILITY SERIES.
MONITOR EDITOR O.S. VI.2
(C) 1985 S.D.C.

followed by a "   " prompt and a flashing cursor. You are now in Monitor command mode. It is from this that you will command most of the functions of the Monitor/Editor and the other programs.

A summary of commands is given later but for now try the following commands.

Type DERR followed by return. The disk status is given on screen in a form similar to:

DISK DR. :0, OK,0,0

where DISK DR. indicates that this is a disk status report. Machine reports are prefixed by MACHINE.

Now type DIR. The directory of the current disk will be displayed on screen in a format similar to that obtained by using the more usual Basic method LOAD"$",8.

Note that some commands, such as COPY and RENAME automatically produce a disk status report to show whether or not the command has been carried out successfully.

The formats of the disk commands SCRATCH, COPY etc. are the same as are given in the 1541 disk users manual. Note that in all cases, spaces, and not quotes, are required around filenames. For example, the command to save a block of memory to disk is S. Thus the format is:

S filename addr1,addr2
and not
S "filename" addr,addr2

Now type I. This will perform a warm start into the monitor. If you type X, the machine will re-enter BASIC. To get back into the Monitor, simply type the SYS 33280 command again.

One thing to note is that the 6502 BRK vector is now set to jump into the monitor. This means that you may insert BRK instructions into your machine code programs to act as STOPs for debugging purposes. To see the effect of this, type the following:

POKE2,0:SYS 2

A zero byte is the code for the 6502 BRK instruction. You will be given the entry message as before but this time it will be preceded by:

BREAK

indicating that a BRK instruction was encountered during execution of a machine code program.

Any other words typed in are assumed to be "external" commands i.e. disk files which act as commands. The programs MACRO and ASSEMBLER are examples of these. When you want to call the Macro Processor for example, assuming you called the program MACRO, you simply type MACRO plus a couple of filenames and the Macro processor will be loaded and run. An external command is assumed to begin at address 9000 hex. Therefore, if you want to write a program to act as an external command, it should have a start address of 9000 hex. At the end of execution, a JMP $8200 instruction will cause a re-entry to the monitor.

Editor Commands

Command	Description
A	Auto line numbering this is a useful feature since it does away with the need to type in line numbers. Two formats are valid. 1 A : start 10, increment 10 2 A10,20 : start 10, increment 20 Format 2 may of course use other numbers.
D	Delete lines there are two methods of line deletion: 1 The usual Basic method, type a line number. 2 The D command. An example would be D20-60 which would delete lines 20 through to 60.
R	Renumber lines this simply renumbers the lines of a source code program. The format is the the same as for the A command.
L	List program lines – the RUN/STOP key may be used to terminate the listing. If a line number is supplied, listing will commence from that line e.g. L200 will begin listing at line 200.
M	Merge file – the format is Mfilename. The file named will be merged onto the end of the file currently in memory.
S	Scratch (delete) a file – this is similar to the Basic command NEW.
I	Initialise – re-enter editor deleting any file currently in memory.
*	Exit editor – there are three formats for this command.
1 * : save file	and exit.
2 *! : save file	without save.
3 *filename : save file	with a new name and exit.

The Editor

The Editor is provided so as to allow preparation of source code in the correct format for the Macro Processor and Assembler. The basic idea behind the Editor screen format is that source code symbols, operands and mnemonics all have defined areas in a source line. These areas are known as FIELDS and both the Macro Processor and Assembler expect to find their source code arranged in these fields.

On a normal C64 screen, there are 25 rows of 40 characters. The editor arranges the screen such that there are four fields of 10 characters each numbered one to four starting at the left side. Since the normal screen editor will accept up to 80 characters i.e. two lines of text, the last field-field four – may extend for up to a maximum of 50 characters. Field one is only used by the Editor to accept and display line numbers. It is not actually stored in the final source program. Field two is the SYMBOL or LABEL field, field three is the INSTRUCTION or MNEMONIC field and field four is the OPERAND field.

The functions of these fields will become clearer as we go on to explain the Macro Processor and Assembler.

A typical Editor screen layout is shown below. The top line is for guidance only and is not on screen.

```
FIELD 1 FIELD 2 FIELD 3 FIELD 4
1000    LABEL 1 JSR    DELAY
1010            LDA    $40
1020            ASL    A
etc.....
```

If you're wondering how you are going to keep all the field entries in the right places then read on. When in edit mode, the function key F1 is used as a TAB key. When pressed, it will advance the cursor from its present position to the start of the next field. For example, suppose the line 1000 in the above example was being typed in. The line number would be typed in and the F1 key pressed. The cursor would be advanced to the start of field 2 – the L of LABEL1. Pressing F1 again would advance the cursor to the start of field 3 – the J of JSR and so on.

This feature coupled with an auto line numbering facility makes for easy code entry.

Other facilities include line renumbering, block line delete, source code file merging and single line delete. A program is entered in a similar fashion to a Basic program with the line numbers. These numbers are for editing purposes only and do not affect the final object code generated by the assembler. All the normal Commodore screen editing facilities are supported by this editor.

Entering and Leaving the Editor

While in Monitor command mode, enter the command EDITOR. The message EDITOR. ENTER FILENAME will appear. You should enter the name of the file you wish to use. If the file exists on the current disk the Editor will read it in and you will be able to work on it. If however the file does not exist on the current disk, the messge

NEW FILE will be printed. In both cases, a flashing cursor (no prompt) will signify Editor command mode.

Before going on to explain the Editor commands, a word or two about leaving it. The name entered as the filename, is stored by the editor. At the end of the editing session the user may simply enter the command * and the file is saved back to disk. An auto disk validate is carried out.

This is done to prevent problems associated with the 1541 " 0:" bug. In some cases it may take a long time to validate a disk especially if it is getting dull. Some of you may find this a bit of a nuisance but I personally prefer it to a corrupted disk. If you want to do away with this facility, refer to listing seven which contains details on removing it.
(see Editor Commands).

When you have the Macro Processor and the Assembler complete, I will give some programming examples to let you get used to the operation of the Editor and its formats. Next month I will give listings of the Macro Processor and a description of Macros and Macro processing.

Monitor Part 1

```
5 REM ** JUMP TABLE **
6 REM ****************
10 DATA 76,114,137,76,152,135,76,1
31,135,76,189,134,76,139,132,76
20 DATA 156,132,76,94,132,76,51,13
2,76,17,132,76,35,132,76,45
30 DATA 132,76,109,135,76,18,135,7
6,16,141
34 REM ****************************
****
35 REM ** COMMAND & MESSAGE TABLES
 **
36 REM ****************************
****
40 DATA 77,0,82,0,71,0,84,0,70,0,7
6,0,83,0,43,0,42,0,73,0,88,0
50 DATA 82,69,83,69,84,0,68,73,82,
0,67,65,84,0,83,67
60 DATA 82,65,84,67,72,0,70,79,82,
77,65,84,0,73,78,73
70 DATA 84,0,67,79,80,89,0,82,69,7
8,65,77,69,0,84,89
80 DATA 80,69,0,67,79,77,80,65,67,
84,0,68,69,82,82,0
90 DATA 72,69,88,0,68,69,67,0,69,6
8,73,84,79,82,0,86
100 DATA 68,85,0,84,84,89,0,255,83
,133,45,133,114,134,50,134
110 DATA 248,133,6,130,3,130,208,1
33,171,133,114,137,158,134,226,252
120 DATA 189,136,189,136,57,135,60
,135,101,135,63,135,66,135,46,136
130 DATA 98,135,9,133,223,135,226,
135,91,139,150,136,156,136,13,10
140 DATA 67,79,77,77,79,68,79,82,6
9,32,54,52,32,85,84,73
150 DATA 76,73,84,89,32,83,69,82,7
3,69,83,46,13,10,77,79
```


Monitor Part 1

```
160 DATA 78,73,84,79,82,32,69,68,7
3,84,79,82,32,79,46,83
170 DATA 46,32,86,49,46,50,13,10,4
0,67,41,32,49,57,56,53
180 DATA 32,83,46,68,46,67,46,13,1
0,13,10,0,13,13,10,42
190 DATA 42,32,66,82,69,65,75,32,6
9,78,84,82,89,32,42,42
200 DATA 13,10,0,42,79,75,46,13,10
,0,13,10,78,69,87,32
210 DATA 70,73,76,69,13,10,13,10,0
,13,10,69,68,73,84,79
220 DATA 82,13,10,0,13,10,42,42,69
,82,82,79,82,42,42,13
230 DATA 10,0,13,10,77,65,67,72,73
,78,69,32,58,32,0,68
240 DATA 73,83,75,32,68,82,32,58,3
2,0,13,10,32,32,32,32
250 DATA 80,67,32,32,83,82,32,65,6
7,32,88,82,32,89,82,32
260 DATA 83,80,13,10,62,43,32,0,13
,10,69,78,84,69,82
264 REM ****************************
****
265 REM ** INPUT EVALUATION SUBRT.
 **
266 REM ****************************
****
270 DATA 32,70,73,76,69,78,65,77,6
9,32,0,238,64,48,58,255,32
280 DATA 138,173,76,247,183,201,48
,144,18,201,71,176,14,201,58,176
290 DATA 3,41,15,96,201,65,144,3,2
33,55,96,76,72,178,32,121
300 DATA 0,32,181,131,72,32,115,0,
32,24,130,144,20,104,10,10
310 DATA 10,10,133,2,32,121,0,32,1
81,131,5,2,133,2,76,115
320 DATA 0,104,133,2,96,32,206,131
,165,2,133,20,32,121,0,32
330 DATA 24,130,176,1,96,32,206,13
1,165,20,133,21,165,2,133,20
340 DATA 96,32,27,130,144,1,96,201
,65,144,6,201,71,176,2,56
350 DATA 96,24,96,201,48,144,250,2
01,58,176,246,56,96,32,19,177
360 DATA 144,241,96,169,0,133,20,1
33,21,32,121,0,201,39,208,6
370 DATA 32,115,0,133,20,96,173,17
4,2,240,3,76,245,131,32,121
380 DATA 0,76,175,131
1000 FORS=33280 TO 33875
1010 READA:POKES,A
1020 NEXT
1030 PRINT"FINISHED"
```

Monitor Part 2

```
5 REM ** OUTPUT SUBROUTINES **
6 REM ************************
10 DATA 169,32,76,210,255,169,44,7
6,210,255,169,13,32,210,255,169
20 DATA 10,76,210,255,169,147,76,2
10,255,72,74,74,74,74,32,129
30 DATA 132,32,210,255,104,41,15,3
2,129,132,76,210,255,201,10,144
40 DATA 3,105,54,96,9,48,96,173,17
4,2,240,8,152,32,109,132
50 DATA 138,76,109,132,152,76,205,
189,72,173,174,2,240,4,104,76
60 DATA 109,132,104,134,253,132,25
4,170,169,0,32,205,189,164,254,166
70 DATA 253,96,169,6,160,5,162,15,
140,32,208,141,134,2,142,33,208,96
74 REM ****************************
****
75 REM ** ERROR HANDLING **
76 REM ****************************
****
80 DATA 138,16,3,108,2,3,72,32,190
,138,169,98,160,131
90 DATA 32,30,171,104,10,170,169,1
28,133,157,189,38,163,133,34,189
100 DATA 39,163,133,35,32,204,255,
169,0,133,19,32,220,170,160,0
110 DATA 177,34,72,41,127,32,71,17
1,200,104,16,244,32,122,166,32
120 DATA 18,130,108,2,3,32,18,130,
169,111,160,131,32,30,171,162
130 DATA 15,32,198,255,32,228,255,
32,210,255,201,13,208,246,162,0
140 DATA 32,198,255,32,18,130,108,
2,3
144 REM ***************************
***
145 REM ** MONITOR O.S. SUBRT.
146 REM ***************************
***
149 DATA 169,122,160,131,32,30,171
150 DATA 172,168,2,174,167,2,32,12
,130,162,0,32,89,132,189,169
160 DATA 2,32,15,130,232,224,5,208
,242,32,18,130,108,2,3,32
170 DATA 21,130,165,20,133,251,165
,21,133,252,32,253,174,32,21,130
180 DATA 32,18,130,169,62,32,210,2
55,169,42,32,210,255,32,84,132
190 DATA 164,252,166,251,32,12,130
,160,0,32,89,132,177,251,32,15
200 DATA 130,200,192,8,144,243,32,
225,255,240,22,152,24,101,251,133
210 DATA 251,165,252,105,0,133,252
,197,21,144,197,165,251,197,20,144
220 DATA 191,32,18,130,108,2,3,32,
21,130,165,20,133,251,165,21
230 DATA 133,252,160,0,132,90,32,2
53,174,32,21,130,164,90,165,20
240 DATA 145,251,200,132,90,192,8,
144,235,108,2,3,32,21,130,165
250 DATA 20,166,21,141,167,2,142,1
68,2,162,0,134,253,32,253,174
260 DATA 32,21,130,165,20,166,253,
157,169,2,232,134,253,224,5,144
270 DATA 234,108,2,3,32,21,130,165
,20,166,21,133,251,134,252,32
280 DATA 253,174,32,21,130,165,20,
166,21,133,253,134,254,32,253,174
290 DATA 32,21,130,160,0,165,20,14
5,251,230,251,208,2,230,252,165
300 DATA 251,197,253,208,240,165,2
52,197,254,208,234,108,2,3,32,21
310 DATA 130,165,20,166,21,133,251
,134,252,32,253,174,32,21,130,165
320 DATA 20,166,21,133,253,134,254
,32,253,174,32,21,130,160,0,177
330 DATA 251,145,20,230,251,208,2,
230,252,230,20,208,2,230,21,165
340 DATA 251,197,253,208,234,165,2
52,197,254,208,228,108,2,3,32,121
350 DATA 0,240,13,32,21,130,165,20
,166,21,141,167,2,142,168,2
360 DATA 174,173,2,154,173,168,2,7
2,173,167,2,72,173,169,2,72
370 DATA 173,170,2,174,171,2,172,1
72,2,64,169,139,162,227,141,0
380 DATA 3,142,1,3,169,131,162,164
,141,2,3,142,3,3,174,173
390 DATA 2,154,169,128,133,157,76,
123,227
400 REM ***************************
***
1000 FORS=33876 TO 34492
1010 READA:POKES,A
1020 NEXT
1030 PRINT"FINISHED"
```

```
5 REM *****************************
***
6 REM I/O SUBROUTINES
7 REM *****************************
**
10 DATA 32,121,0,162,0,164,122,132
,93,185,0,2,240,15,201,32
20 DATA 240,11,200,232,224,50,144,
241,162,23,108,0,3,132,122,224
30 DATA 0,208,5,162,8,108,0,3,164,
93,96,32,9,130,134,2
40 DATA 152,170,169,58,202,157,0,2
,202,165,94,157,0,2,134,94
50 DATA 166,2,232,232,134,2,165,94
,133,90,169,2,133,91,164,2
60 DATA 169,0,145,90,96,169,15,162
,8,160,15,32,186,255,169,0
70 DATA 32,189,255,76,192,255,169,
15,76,195,255,162,15,32,201,255
80 DATA 165,94,160,2,32,30,171,162
,0,76,201,255,169,83,44,169
90 DATA 78,44,169,67,44,169,82,133
,94,32,232,134,32,40,135,76
100 DATA 9,133,141,70,2,169,0,141,
71,2,169,1,133,2,169,70
110 DATA 133,94,76,40,135,169,86,4
4,169,73,32,79,135,108,2,3
120 DATA 32,9,130,138,72,152,170,1
04,160,2,76,189,255,169,1,162
130 DATA 8,160,1,76,186,255,32,33,
130,32,122,135,169,0,32,213
140 DATA 255,144,5,162,4,108,0,3,1
08,2,3,32,33,130,32,122
150 DATA 135,169,32,32,255,174,32,
21,130,165,20,166,21,72,138,72
160 DATA 32,253,174,32,21,130,166,
20,164,21,104,133,21,104,133,20
170 DATA 165,1,41,254,133,1,169,20
,32,216,255,8,165,1,9,1
180 DATA 133,1,40,144,5,162,24,108
,0,3,32,183,255,208,246,108
190 DATA 2,3,169,255,44,169,0,141,
174,2,169,51,160,131,32,30
200 DATA 171,108,2,3,32,207,255,72
,32,183,255,41,64,208,2,104
210 DATA 96,104,162,0,32,198,255,1
69,2,32,195,255,108,2,3,169
220 DATA 2,162,8,160,2,32,186,255,
32,33,130,24,32,192,255,176
230 DATA 11,32,183,255,208,6,96,16
2,2,76,198,255,162,4,108,0
240 DATA 3,32,12,136,32,204,255,32
,131,136,32,241,135,32,241,135
250 DATA 32,241,135,32,241,135,32,
241,135,32,241,135,160,0,32,241
260 DATA 135,48,251,153,0,2,200,20
1,0,208,243,32,204,255,32,120
270 DATA 136,169,0,160,2,32,30,171
,32,18,130,32,204,255,32,131
280 DATA 136,32,225,255,240,140,16
0,0,76,61,136,173,175,2,208,1
290 DATA 96,162,200,76,201,255,162
,2,76,198,255,162,0,142,175,2
300 DATA 32,201,255,169,200,32,195
,255,96,32,136,136,108,2,3
310 DATA 169,200,141,175,2,162
311 REM ***************************
****
312 REM THIS IS THE DEVICE NUMBER
FOR
313 REM THE PRINTER
314 REM ***
315 DATA 4
316 REM ***
317 REM CHANGE TO WHATEVER YOU NEE
D
318 REM ***************************
****
319 DATA 160,1,32,186,255,169,0,32
,189,255
320 DATA 169,4,141,147,2,32,192,25
5,162,200,32,201,255,108,2,3
330 DATA 32,136,136,169,36,133,251
,169,48,133,252,169,2,162,251,160
340 DATA 0,32,189,255,169,1,162,8,
160,0,32,186,255,32,192,255
350 DATA 144,10,72,165,184,32,195,
255,104,76,9,133,160,3,132,183
360 DATA 166,184,32,198,255,32,207
,255,133,87,32,183,255,208,105,32
370 DATA 207,255,133,88,32,183,255
,208,95,164,183,136,208,224,132,18
3
380 DATA 32,207,255,72,32,183,255,
170,104,224,0,208,75,164,183,192
390 DATA 80,176,69,153,0,2,170,240
,4,230,183,208,227,162,0,32
400 DATA 198,255,166,87,165,88,32,
205,189,169,32,32,210,255,160,0
410 DATA 185,0,2,240,6,32,210,255,
200,208,245,32,18,130,162,0
420 DATA 32,198,255,32,225,255,240
,16,32,228,255,201,32,208,5,32
430 DATA 228,255,240,251,160,2,208
,164,162,0,32,198,255,165,184,32
440 DATA 195,255,76,9,133
```

Monitor Part 3

```
444 REM ***************************
***
445 REM ENTRY POINT/MAIN LOOP
446 REM ***************************
***
449 DATA 169,150,162,137,141,22,3,
142,23,3,169
450 DATA 198,162,132,141,0,3,142,1
,3,169,219,162,137,141,2,3
460 DATA 142,3,3,234,234,234,76,18
8,137,104,141,172,2,104,141,171
470 DATA 2,104,141,170,2,104,141,1
69,2,104,105,255,141,167,2,104
480 DATA 105,255,141,168,2,169,28,
160,131,32,30,171,76,114,137,32
490 DATA 182,132,169,1,141,174,2,1
69,130,162,0,134,51,134,55,133
500 DATA 52,133,56,169,206,160,130
,32,30,171,186,142,173,2,32,182
510 DATA 132,169,0,133,157,32,35,1
35,169,62,32,210,255,174,173,2
520 DATA 154,32,96,165,134,122,132
,123,32,115,0,201,62,240,249,201
530 DATA 0,240,219,166,122,134.93.
160,0,132,2,185,42,130,201,255
540 DATA 240,65,201,0,240,25,221,0
,2,208,4,232,200,208,236,185
550 DATA 42,130,240,3,200,208,248,
200,166,122,230,2,76,8,138,189
560 DATA 0,2,32,19,177,144,6,185,4
2,130,76,19,138,165,2,10
570 DATA 134,122,170,189,152,130,1
33,20,189,153,130,133,21,32,18,135
580 DATA 108,20,0,165,93,133,122,1
69,2,133,123,32,18,135,32,33
590 DATA 130,32,122,135,24,169,0,3
2,213,255,176,6,32,35,135,76
600 DATA 0,144,162,4,108,0,3
1000 FOR S=34493 TO 35443
1010 READ A:POKES,A
1020 NEXT
1030 PRINT"FINISHED"
```

Monitor Part 4

```
5 REM ******************************
***
6 REM EDITOR SUBROUTINES 1
7 REM ******************************
***
10 DATA 162,10,160,4,189,0,2,232,2
00,153,251,1,185,251,1,208
20 DATA 243,153,253,1,198,123,169,
255,133,122,96,120,169,116,162,138
30 DATA 141,4,3,142,5,3,169,240,16
2,138,141,2,3,142,3,3
40 DATA 169,255,133,129,169,104,16
2,140,141,20,3,142,21,3,169,218
50 DATA 162,141,141,0,3,142,1,3,88
,96,120,169,124,162,165,141
60 DATA 4,3,142,5,3,169,32,133,129
,169,219,162,137,141,2,3
70 DATA 142,3,3,169,49,162,234,141
,20,3,142,21,3,169,198,162
80 DATA 132,141,0,3,142,1,3,88,96
84 REM ******************************
****
85 REM ** EDITOR MAIN LOOP
86 REM ******************************
****
89 DATA 108,2,3,32,35,135,32
90 DATA 18,135,32,96,165,134,122,1
32,123,32,115,0,170,240,237,162
100 DATA 255,134,58,144,68,201,42,
208,3,76,252,139,201,76,208,3
110 DATA 76,88,141,201,82,208,3,76
,241,140,201,73,208,6,32,190
120 DATA 138,76,91,139,201,68,208,
3,76,166,140,201,83,208,6,32
130 DATA 80,140,108,2,3,201,77,208
,3,76,178,141,201,65,208,6
140 DATA 32,228,141,108,2,3,108,0,
3,32,107,169,32,121,0,208
150 DATA 3,76,230,140,76,159,164,3
2,136,136,169,73,160,131,32,30
160 DATA 171,32,80,140,169,0,141,1
74,2,169,152,160,131,32,30,171
170 DATA 32,96,165,134,122,132,123
,32,115,0,170,208,3,76,109,139
180 DATA 160,0,185,170,131,153,176
,2,200,201,255,208,245,32,33,130
190 DATA 165,183,201,20,144,5,162,
23,108,0,3,160,0,162,3,177
200 DATA 187,153,180,2,200,232,196
,183,144,245,142,176,2,169,180,160
210 DATA 2,133,187,132,188,202,202
,202,134,183,32,122,135,32,143,138
220 DATA 169,0,32,213,255,144,13,3
2,18,130,169,58,160,131,32,30
```

Monitor Part 4

```
230 DATA 171,76,101,135,32,18,130,
32,228,139,32,99,166,108,2,3
240 DATA 169,255,160,1,145,43,32,5
1,165,165,34,24,216,105,2,133
250 DATA 45,165,35,105,0,133,46,96
,160,1,177,43,208,3,76,66
260 DATA 140,32,228,139,32,115,0,2
40,13,201,33,240,49,32,33,130
270 DATA 32,122,135,76,39,140,32,1
22,135,173,176,2,162,177,160,2
280 DATA 32,189,255,166,45,164,46,
169,43,32,216,255,144,3,76,210
290 DATA 135,32,183,255,240,3,76,2
10,135
291 REM **************************
***
292 REM TO REMOVE AUTO-VALIDATE, R
EPLACE LINE 295 WITH THIS
293 REM 295 DATA 234,234,234,234,2
34
294 REM **************************
***
295 DATA 169,86,32,79,135
296 REM **************************
****
300 DATA 32,190,138,169,1,141,174,
2,32,18,130,76,114,137
304 REM **************************
****
305 REM ** EDITOR SUBROUTINES 2
306 REM **************************
****
310 DATA 169,0,168,145,43,200,145,
43,165,43,24,105,2,133,45,165,44,1
05,0,133
320 DATA 46,76,99,166,165,197,201,
64,208,3,76,49,234,201,4,208
330 DATA 249,165,198,208,245,56,32
,240,255,152,56,233,10,176,252,73
340 DATA 255,105,1,170,232,160,0,2
02,240,9,169,29,153,119,2,200
350 DATA 76,139,140,132,198,162,96
,160,255,136,208,253,202,208,248,7
6
360 DATA 49,234,32,115,0,32,21,130
,32,19,166,144,61,165,95,72
370 DATA 165,96,72,169,45,32,255,1
74,32,21,130,32,19,166,144,42
380 DATA 160,1,177,95,170,136,177,
95,168,104,133,96,104,133,95,152
390 DATA 160,0,145,95,200,138,145,
95,200,177,95,133,20,200,177,95
400 DATA 133,21,169,0,141,0,2,76,1
64,164,76,227,168,32,247,140
410 DATA 108,2,3,32,115,0,240,20,3
2,21,130,165,20,133,251,165
420 DATA 21,133,252,32,253,174,32,
21,130,76,28,141,169,10,162,0
430 DATA 133,251,133,20,134,252,13
4,21,165,43,166,44,133,253,134,254
440 DATA 160,1,177,253,208,4,32,18
,130,96,160,3,165,252,145,253
450 DATA 136,165,251,145,253,136,1
77,253,170,136,177,253,133,253,134
,254
460 DATA 165,251,24,101,20,133,251
,165,252,101,21,133,252,80,209,162
470 DATA 15,108,0,3,32,115,0,240,9
,32,21,130,32,19,166,76
480 DATA 110,141,165,43,166,44,133
,95,134,96,160,1,177,95,208,6
490 DATA 32,18,130,108,2,3,160,2,1
77,95,170,200,177,95,32,205
500 DATA 189,56,32,240,255,160,10,
24,32,240,255,160,4,177,95,240
510 DATA 6,32,210,255,200,208,246,
32,18,130,160,1,177,95,170,136
520 DATA 177,95,133,95,134,96,32,2
25,255,240,197,76,110,141,32,115
530 DATA 0,32,33,130,169,1,162,8,1
60,0,32,186,255,165,45,56
540 DATA 233,2,170,165,46,233,0,16
8,169,0,24,32,213,255,176,3
550 DATA 76,216,139,108,0,3,169,84
,160,131,32,30,171,108,2,3
560 DATA 32,115,0,208,13,169,10,16
2,0,133,251,134,252,133,253,76
570 DATA 11,142,32,21,130,32,253,1
74,165,20,133,251,165,21,133,252
580 DATA 32,21,130,165,20,133,253,
169,21,162,142,141,2,3,142,3
590 DATA 3,173,0,2,240,46,166,251,
165,252,32,85,142,169,11,133
600 DATA 198,160,0,169,29,153,119,
2,200,192,11,144,246,189,0,2
610 DATA 157,119,2,202,16,247,24,1
65,251,101,253,133,251,144,2,230
620 DATA 252,76,240,138,169,240,16
2,138,141,2,3,142,3,3,108,2
630 DATA 3,134,99,133,98,162,144,5
6,32,73,188,32,223,189,32,135
640 DATA 180,32,166,182,162,0,189,
0,1,157,0,2,240,3,232,208
650 DATA 245,96
1000 FOR S=35444TO36469
1010 READA:POKES,A
1020 NEXT
1030 PRINT"FINISHED"
```

THE 64 SOFTWARE CENTRE
1 Princeton Street, London WC1R 4AL
01-430 0954

SOFTWARE PRICE LIST — Prices include VAT

d = disk c = cassette r = cartridge

Accounting systems (Office use)

Anagram Systems — Sales Ledger	d	£75.00
Anagram Systems — Purchase/Nominal Ledger	d	£75.00
Anagram Systems — Cashbook	d	£75.00
Gemini — Cashbook (with nominal analysis)	d	£64.95
Gemini — Cashbook (with nominal analysis)	c	£59.95
Gemini — Final Accounts	d	£64.95
Gemini — Final Accounts	c	£59.95
Gemini — Cashbook/VAT/Final Accounts	c	£89.95
Gemini — Cashbook/VAT/Final Accounts	d	£99.95
Ramtop — Accounts Package	d	£75.00
Studio — Payroll 64	d	£49.95
Dell — Invoice Generator II	d	£24.95

Stock Control systems

Practicorp — Inventory 64	d	£29.95
Gemini — Stock Control	d	£24.95
Gemini — Stock Control	c	£19.95
Anagram — Stock Control	d	£75.00

Home applications

Gemini — Home Accounts	c	£19.95
Gemini — Home Accounts	d	£24.95
Gemini — Combined Home Pack	d	£39.95
Gemini — Combined Home Pack	c	£34.95
Fieldmaster — Home accounts	d or c	£19.95
Adamsoft — Checkbook Manager	d	£14.95

Database Filing Systems

Handic — Diary 64	r	£19.95
Fieldmaster — Mail label	d	£29.95
BI — The Consultant	d	£125.00
Practicorp — Practifile 64	d	£44.50
Precision — Superbase 64	d	£87.00
Precision — Superbase Starter	d	£39.95
Gemini — Database	d	£24.95
Gemini — Database	c	£19.95
Gemini — Mailist	d	£24.95
Gemini — Mailist	c	£19.95
Fieldmaster — Record Card	d or c	£19.95
Dell — Superfile	d	£14.95
Practicorp — Practicalc 64	d	£44.50
Practicorp — Practicalc II	d	£69.95
Supersoft — Busicalc 1	d or c	£17.95
Supersoft — Busicalc 3	d	£49.95
Fieldmaster — Worksheet	d, f or c	£19.95

Spreadsheets

Handic — Easy Calcresult	r	£49.95
Handic — Advanced Calcresult	r + d	£90.00

Word Processors

B.I. — Paperclip 64	d	£90.00
Simple — Simply Write	d	£46.00
Simple — Simply Write	c	£40.25
Hesware — Heswriter	r	£42.95
Fieldmaster — Pagewriter	d or c	£29.95
Bank Street Writer	d	£69.00
Precision — Superscript	d	£69.95
DTL — Wordcraft 40	r	£69.95

Utilities

Adamsoft — Ultrabasic	d	£19.95
Adamsoft — Cadpack 64	d	£35.00
Adamsoft — Chartpak 64	d	£24.95
Adamsoft — Superdisk	d	£12.50
Adamsoft — Graphics Designer	d	£19.95
Audiogenic — Forth 64	r	£29.95
Blitz compiler	d	£50.00
Fieldmaster — Poster printer	c	£19.95
Handic — Mon 64	r	£34.95
Hesware — Forth 64	r	£59.95
Commodore — Simons Basic	r	£47.50
Kuma — BC Basic	r	£57.50
Practicorp — 64 Doctor	d	£19.95
Supersoft — Master 64	d	£71.30
Supersoft — Victree	r	£56.35
Supersoft — Mikro Assembler	r	£59.80
Supersoft — Zoom monitor	d	£14.95
Supersoft — Toolkit 64	c	£9.95
Supersoft — Toolkit 64	d	£11.95
Aztec — Turbo extended basic	c	£14.95
Activision — The Designer's Pencil	c	£11.99
Jetpack compiler	d	£39.95
Jetpack compiler	c	£14.95
FP — Pascal 64	d	£34.95
Oasis — Basic Lightning	c	£14.95
Oasis — Basic Lightning	d	£19.95
Oasis — White Lightning	c	£19.95
Oasis — White Lightning	d	£29.95
Oasis — Machine Lightning	c	£29.95
Oasis — Machine Lightning	d	£39.95
Talent — Panorama	c	£17.95
Talent — Panorama	d	£19.95
Skyles — 1541 Flash kit	r	£89.95
Epyx — Fast load	r	£49.95
Reston — Moviemaker	d	£65.00
Access — Mach 5 fast loaders	r or d	£39.95
Orpheus Pascal	c	£24.95
Broderbund — The Printshop	d	£44.95
Broderbund — Printshop graphics designer	d	£29.95

Simulations

Flight simulators

Microprose — Solo Flight	c or d	£14.95
Sublogic — Flight Simulator II	d	£49.95
Sublogic — Flight Simulator II	c	£42.95
Supersoft — Interdictor Pilot	d	£19.95
Supersoft — Interdictor Pilot	c	£17.95
Activision — Space Shuttle	c	£11.99
Cosmi — Super Huey (helicopter)	c	£11.95

War

SSI — Cosmic Balance	d	£44.95
SSI — Germany 1985	c or d	£59.95
SSI — Operation Market Garden	d	£64.95
SSI — Battle for Normandy	c or d	£14.95
SSI — Geopolitique	d	£44.95
SSI — Breakthrough in the Ardennes	d	£65.00
SSI — Baltic 85	d	£44.95
SSI — Broadsides	d	£44.95
SSI — Field of Fire	d	£44.95

ALL IN STOCK NOW — Callers welcome
(10am – 6pm including Saturdays)

Mail Order: Cheque or PO — add £2 for items over £20, others free.
Credit cards (Access/Visa): phone or write. Immediate despatch.

The 64 Software Centre,
1 Princeton Street, London WC1

Please supply the following items:

1.................... Qty.... @ £ ....
2.................... Qty.... @ £ ....
3.................... Qty.... @ £ ....
4.................... Qty.... @ £ ....

TOTAL £ ....
POSTAGE £ ....
TOTAL £ ....

Name ....
Visa/Access Card No. ....
Address ....
....
Date .... Signature ....

This month Joe Nicholson continues our C-16 series with an explanation of programmable characters.

PROGRAMMING THE C16

Programmable characters

IT IS POSSIBLE TO USE CUSTOM DESigned character sets on the C-16, even though Basic 3.5 does not have supporting commands.

To understand how this is done, one must first examine how the normal character set is stored. Data for the character shapes is stored in the ROM from address $D000 to address $D7FF (53258 to 55295). Locations $D000 to $D3FF (53248 to 54271) hold the data for character set 1 (upper case letters and graphics). Locations $D400 to $d7FF (54272 to 55295) hold the data for character set 2 (upper and lower case letters). Each of the two character sets takes up 1K of memory. Characters with screen codes between 128 and 255 are reversed images of codes 0 to 127 and are therefore not stored in memory.

The characters are stored in the order shown in the screen display codes in Appendix E of the *C-16 User Manual*. For set 1, the first character is therefore '@'. This is stored in eight bytes, one byte per pixel line (eight dots) of the screen display. Each byte of the character '@' contains the eight bits needed for each row of the character, stored in binary form (one for on, nought for off). The left-most bit of the row is the '128' bit, the second to the left is lthe '64' bit and so on to the '1' bit on the far right.

The '@' sign is therefore stored as shown in Figure 1.

For a character set to be created in RAM, space must first be made available in which to put the character set. Assuming the high-resolution screen will not be used in conjunction with programmable characters, the top 1K of RAM (15360-16383) is the most convenient. This is done by moving the 'Highest address used by Basic' pointer (55-56) and the 'Bottom of string storage' pointer (51-52) down 1K from the top of RAM (see last month's article: The Memory map and where to store machine code). Type:

POKE 56,59:POKE 52,59:CLR

As 'CLR' is used this should be done at the beginning of the program.

Assuming you don't want to redefine all 128 characters of the new character set, you need first to move one of the ROM character sets down into the 1K block. This can be done easily by entering the MONITOR and typing:

T D000 D3FF 3C00 — for set 1, or:
T D400 D7FF 3C00 — for set 2.

Then enter 'X' to leave the Monitor.

To move the character set down inside a Basic program is more difficult. A FOR-NEXT loop takes over 15 seconds, so I've written a short machine code routine which does the task almost instantly. The program is completely relocatable, i.e. it will work wherever it is stored in memory. The Start, End and Length values may be altered as desired for different applications. Figure 2 shows an assembler listing of the routine using the C-16 Assembler published in the June edition of Your Commodore. It is positioned in a free space below Basic at $600 hex, 1536 decimal (see last month: Where to store machine code). To execute the routine from Basic, type: SYS 1536.

To make your character set the current one, you must first disable the Shift+Commodore Key with PRINT CHR$(8), and then set the 'Character data base address' pointer (at 65299) as desired. Bits two to seven of this pointer are the upper six bits of the high byte of the character set address. This enables the character set to start at any multiple of 1K. We are using the fifteenth K (15360 to 16383), so the number entered is 60 (15*4) decimal.

To specify that the character set will be

```
START: 10000  ;SHIFT MEMORY
 10010  ORG $600
 10015  ;
 10020  ;START READING FROM
 10030  LDA #0
 10040  STA $D0
 10050  LDA #$D0
 10060  STA $D1
 10070  ;
 10080  ;START WRITING TO
 10090  LDA #0
 10100  STA $D2
 10110  LDA #3C
 10120  STA $D3
 10130  ;
 10140  ;NUMBER OF BLOCKS TO
 10150  ;BE MOVED (4 FOR 1K)
 10160  LDX #4
 10170  ;
 10180  ;PERFORM MOVE
 10190  :L1 LDY #0
 10200  :L2 LDA ($D0),Y
 10210  STA ($D2),Y
 10220  INY
 10230  BNE R:L2
 10240  INC $D1
 10250  INC $D3
 10260  DEX
 10270  BNE R:L1
 10280  RTS
>> OK.
```

Figure 1.

Address hex	Hex	Binary	Image
$D000	$3C	00111100	****
$D001	$66	01100110	** **
$D002	$6E	01101110	** ***
$D003	$6E	01101110	** ***
$D004	$60	01100000	**
$D005	$62	01100010	** *
$D006	$3C	00111100	****
$D007	$00	00000000	

Figure 2. Shift memory routine

accessed from RAM as opposed to ROM, bit two of address 63298 must be set. As it is important that the other bits at that address should remain unchanged, a line like this should be used:

POKE 65298,PEEK (65298) AND 251

to specify 'character set in RAM'. To get back to the normal ROM character set, the following POKEs should be entered:

POKE 65299, 208
POKE 65298,PEEK (65298) OR 4

Whenever an Error is encountered, the ROM/RAM select bit is reset back to ROM, creating havoc on the screen if the 'Character data base address' pointer is not pointing to the ROM character set. This means that editing should always be done in normal (ROM) character mode.

Also, it is a good idea to put the 'get back to the normal character set' commands as the destination of a TRAP command to stop this happening (see page 141 of the *User Manual*). Remember, however, that the TRAP command must come after the CLR command used when lowering the top of RAM.

```
10 REM PROGRAMMABLE GRAPHICS DEMO
100 POKE56,59:POKE52,59:CLR
110 FORA=1536TO1570:READB:POKEA,B:NEXT:S
YS1536
150 DATA169,0,133,208,169,208,133,209,16
9,0,133,210
160 DATA169,60,133,211,162,4,160,0,177,2
08,145,210
170 DATA200,208,249,230,209,230,211,202,
208,240,96
200 RESTORE10000:FORA=0TO79:READB:POKE15
360+A,B:NEXT
210 PRINTCHR$(8):POKE65299,60:POKE65298,
PEEK(65298)AND251
220 TRAP250:GOTO1000
250 PRINTERR$(ER),EL
260 POKE65299,208:POKE65298,PEEK(65298)O
R4:END
1000 COLOR0,1:COLOR4,1:COLOR1,8,6:SCNCLR
1010 PRINT"FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFF";
1020 FORA=1TO22:PRINT"F"TAB(39)"F";:NEXT
1030 PRINT"FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFF"
1040 COLOR1,2,6:CHAR,18,21,"GHI"
1050 COLOR1,2,4:CHAR,0,24," 0015423
                        0000000"
1100 FORB=3TO12:FORA=1TO6:COLOR1,A+1,4
1110 CHAR,B,(A*3)-1," @AB   @AB   @AB   @AB
   @AB"
1120 CHAR,B,(A*3)," CDE   CDE   CDE   CDE
CDE"
1130 FORC=1TO100:NEXT:NEXT:NEXT
1200 FORB=12TO3STEP-1:FORA=1TO6:COLOR1,A
+1,4
1210 CHAR,B,(A*3)-1,"@AB   @AB   @AB   @AB
  @AB "
1220 CHAR,B,(A*3),"CDE   CDE   CDE   CDE   C
DE "
1230 FORC=1TO100:NEXT:NEXT:NEXT
1240 GOTO1100
10000 DATA32,96,192,159,191,255,255,253
10010 DATA0,0,0,195,231,231,255,61
10020 DATA4,6,3,249,253,255,255,63
10030 DATA253,252,255,255,121,63,1,1
10040 DATA189,24,255,60,231,255,195,195
10050 DATA191,63,255,255,158,252,128,128
10060 DATA255,128,191,161,165,189,129,25
5
10070 DATA0,0,0,3,15,31,63,255
10080 DATA24,24,126,255,255,255,255,255
10090 DATA0,0,0,192,240,248,252,255
```

Figure 3. Programmable graphics demo

Entering programmable characters

The address of the character in RAM can be found as follows:

Address = Base address + (screen code * 8)

The eight bytes for each character can be read into memory by a simple FOR-NEXT loop, with the numbers stored in DATA statements. Figure 3 is a demonstration program to illustrate the points covered in this article. The program functions as follows:

LINE 100 shifts the top of memory down 1K for the new character set.
LINE 110 POKEs into memory the machine code routine SHIFT-MEM held in the DATA statements in line 150-170. It then calls this routine. The routine shifts the ROM upper case character set down into the new RAM area.
LINE 200 READs in the Programmable character data stored in lines 10000-10090 into the start of the new character set (first character to be defined is '@', then 'A', then 'BN' etc.).
LINE 210 – 'PRINT CHR$(8)' – disables the ability to change character sets with the Shift+Commodore Key. As only one character set has been redefined, this disables the ability to change to a garbage character set.
LINE 220 turns on TRAP mode to Line 250 and jumps to the Demonstration (Line 1000).
LINE 250-260 The TRAP routine.
LINE 250 prints the error and the line number on which the error occurred.
LINE 260 first changes the BASE address to point to the ROM character set, then selects 'character set to be taken from ROM', then ENDS.
LINE 1000 defines the colours and clears the screen.
LINEs 1010-1030 print the border.
LINE 1040 prints your base.
LINE 1050 prints the scores.
LINEs 1100-1130 move aliens right.
LINEs 1200-1230 move aliens right.
LINE 1240 performs this spectacular feat of imagination all over again.
LINEs 10000-10090 the programmable characters.
LINEs 10000-10050 the six programmable characters needed for the alien.
LINE 10060 the border character.
LINE 10070-10090 the base.

LISTINGS

Listings will be much easier to enter with our new system.

COMMODORE LISTINGS ARE RATHER well known for the horrible little black blobs that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to find on the Commodore keyboard.

For this reason Your Commodore started to precede any control characters with a REM statement on the previous line that explained exactly what the black blobs were meant to be. Unfortunately the graphics characters were not documented and these still cause some confusion. For this reason we are starting to use a new method for marking the control and graphic characters in our listings.

In future all control and graphics commands will be replaced by mnemonic within square brackets. This mnemonic is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.

Any character that is accessed by pressing shift and letter will be printed as [s LETTER]

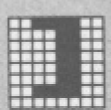
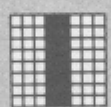
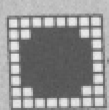
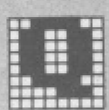
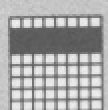
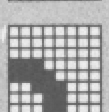
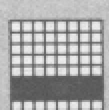
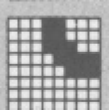
[s A]	shift and A
[s C]	shift & C

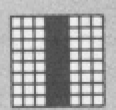
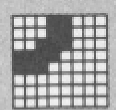
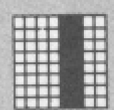
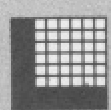
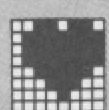
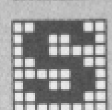
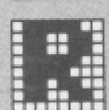
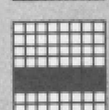
Any character that is accessed by pressing the Commodore key and a letter will be printed as [c LETTER]

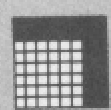
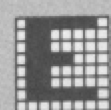
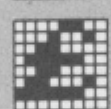
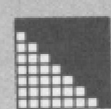
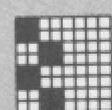
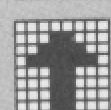
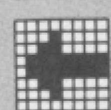
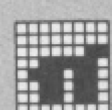
[c A]	Commodore & A
[c C]	Commodore & C
[c 1]	Commodore & 1

Any control key will be printed out as a number. For example [001]. Control codes are accessed by pressing the CTRL and a letter at the same time [001] is CTRL & A, 002 is CTRL & B etc. See the manual for more information about control codes.

[001]	CTRL & A
[026]	CTRL & Z

Mnemonic	Symbol	what to press
[RIGHT]		left/right
[LEFT]		shift left/right
[UP]		Shift & up /down
[DOWN]		up/down
[F1]		f1
[F2]		shift & f1
[F3]		f3
[F4]		shift & f3

Mnemonic	Symbol	what to press
[F5]		f5
[F6]		shift & f5
[F7]		f7
[F8]		shift & f7
[CLEAR]		shift & CLR /HOME
[HOME]		CLR/HOME
[RVSON]		CTRL & 9
[RVSOFF]		CTRL & 0

Mnemonic	Symbol	what to press
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written an exhilirating game or an invaluable utility on your Commodore micro, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be documented and type-written and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original; if it is not chosen for publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below — you never know, you might even get paid for airing your views! All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications Limited
No 1 Golden Square
London WIR 3AB

★PLEASE COMPLETE IN BLOCK CAPITALS

Your Name ____________________

Program Name ____________________

Computer/memory size it runs on ____________________

Amount of memory program occupies ____________________

Other computers/memory size which your program runs on without conversion or use ____________________

Does your game need or use joysticks? Yes No

Have you sent your game to another magazine Yes No

Is it original/or a variation on a theme? ____________________

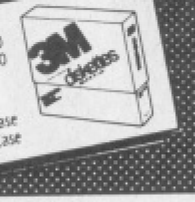
Your Address ____________________

Telephone Number ____________________

Times to contact you ____________________

Evesham Micros
THE UTILITY SPECIALISTS

3M Floppy Disks



BOXES OF TEN DISCS
SS/DD £16.00
DS/DD £20.00
NEW SPECIAL PACKS WITH PLASTIC CASE
SS/DD .... £17.00 including Case
DS/DD .... £21.00 including Case



DATA RECORDER
A **dedicated** data recorder designed for use on the CBM 64 or Vic-20. Specification as C2N but **with pause button.** Model available for the PLUS4 or 16 at same price.
SATISFACTION GUARANTEED.
ONLY £29.95

HARDWARE SPECIALS

STAR SG10C
The ultimate Commododre ready printer. 120 cps or 40 cps in near letter quality mode. Friction and tractor feed.
ONLY £249.00

MPS801 PRINTER
Popular printer at a stunning price.
ONLY £99.95

MPS803 PRINTER
Exceptional print quality for the price.
ONLY £139.95

1541 DISC DRIVE
Look at this price, it is not a misprint.
ONLY £99.95

DISC NOTCHER
Handy new tool allows you to punch a second write protect notch in your discs. Double your disc capacity by using both sides of your discs.
ONLY £9.95

FREEZE FRAME

THE NEW GENERATION OF BACKUP METHODS HAS ARRIVED

"Freeze Frame" is the product that '64 owners have been waiting for. At the flick of a switch **"Freeze Frame"** will take control of your computer and freeze the program in memory, allowing you to **SAVE it to tape or disc**. The uses are endless, but typical applications are . . .

TAPE TO DISC: ANY program can be frozen and **transferred to disc** regardless of the type of loading technique used. (Disc save incorporates a fast loading system.)

DISC TO TAPE: ANY memory resident disc program can be backed up onto tape. (Choice of normal or turbo save speed.)

DISC TO DISC: It is possible to make a self contained back up of disc based programs that will both **load at high speed** and **allow the rest of the disc to be used as normal**.

TAPE TO TAPE: Tape back ups can be made at turbo speed with just one tape deck.

In fact **"Freeze Frame" will FREEZE any program** that is memory resident, and allow you to make a back up of that program onto tape or disc. The beauty being that the saved version can be a fast booting single file. The process is **completely automatic**, no knowledge of BASIC or machine language is required, just follow the prompts and the simple instructions for **100% success**.

"Freeze Frame" is a hardware "device" that plugs into the cartridge port of the '64. It does **NOT** dump the entire contents of memory, just the **working program**. Programs converted run **independently** of the hardware.

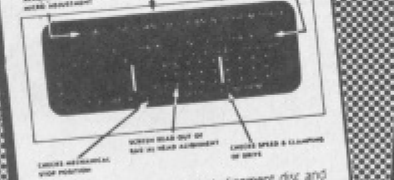
ONLY £39.95

WARNING: Don't be mislead by other adverts, if you want to transfer turbo load programs to disc, software only utilities are **inadequate**. They suffer from a variety of drawbacks including poor success rate, awkward to use, greedy on disc space and overpricing.

ALIGNMENT PROBLEMS?

1541 PHYSICAL EXAM £39.95

IS YOUR 1541 HEALTHY? OR WOULD ITS PHYSICAL EXAM LOOK LIKE THIS ONE?



The program includes a digital alignment disc and software which allows you to **accurately check and correct** disc drive alignment. The instruction booklet is **very detailed**. Also includes quiet drive stops to **stop** that "hammering" thus preventing further problems.

ONLY £39.95

DISC DISECTOR V3.0

Now the ultimate disc copying and utility program for the CBM 64 with 1541 disc drive has been **improved even further**. As well as general improvements plus the addition of some useful utilities two important new programs have been added.

The Evesham Nibbler
This is the first copier of its kind to appear on the U.K. market. It is a **true nibbler** (byte for byte) copier that will copy **ALL DOS errors automatically**. This includes DOS errors 20-29 inc. plus non standard errors like half tracks, extra tracks, non standard sync, renumbered tracks, etc. They are all treated as normal work. No knowledge is required at all it just **"nibbles away"** and produces a perfect copy. Yet it **only takes eight minutes**. Has copied **all discs tested** (except itself). A **class above** anything else available.

Fast File Copier
An **incredibly handy** selective file copier that LOADs and SAVEs at **five times** normal speed. Sounds useful? You won't believe how much until you have used it.

As well as these programs "Disc Disector" incorporates a whole host of useful utilities including the following. **Fast Format**, **Selective Menu Maker**, **Disc Editor** (new sophisticated machine code version), **Scratch/Unscratch**, **Fastcopy**, **Rename**, **Fastload** (four times faster), etc. etc.
This program is an **essential purchase** for the 1541 user.

ONLY £29.95

Customers with earlier versions may return them along with £9.95 for "V3.0".

Quickdisc+
FAST LOAD PLUS UTILITY CARTRIDGE

Get a **QUICKDISC+** cartridge plugged into your '64 and your 1541 will really **start to perform**. So many features per pound (sterling) that you will hardly believe it. Includes the following . . .
Fast LOAD and SAVE (four to five times normal speed). Works with most protected software. Can be switched in and out from the keyboard.
Fast Format takes just 10 seconds.
Fast Backup copies an entire disc in four minutes. (Not heavily protected software.)
Fast File Copier for selective fast copying.
Improved DOS commands (DOS 5.1) makes for easy use of the disc drive. e.g. $(RETURN) will LOAD and display a directory without overwriting BASIC. SHIFT RUN/STOP will LOAD"0:*",8,1 etc. etc. Very, very useful.
Incorporates Centronics printer software (user port) **with CBM graphics** capability. See "Commodore Connexion" for suitable lead.
A RESET switch is fitted. (We have found this to be "unstoppable", it also preserves the tape buffer.)
NO MEMORY IS USED by this cartridge, it is totally "transparent" and uses special switching techniques.
Now the price. No, not £49.95, it's **not even half** that much. Can you afford to be without QUICKDISC+.

ONLY £19.95

No extra connections are necessary with QUICKDISC+ and unlike most similar products it is compatible with printers and second drives.

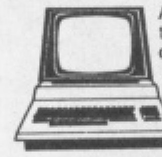
All prices include P&P and VAT. Send cheque, Postal Order of Credit Card No. Credit Card order accepted by phone. Callers welcome. Wide range of goods stocked. Trade enquiries welcome. European orders send price as advertised. Outside Europe £2.00 for airmail. Mail order to Evesham Shop please.

EVESHAM MICROS
BRIDGE STREET, EVESHAM, WORCS., WR11 4RY.
Tel: 0386 41989

MICRO CENTRE
1756 PERSHORE ROAD, COTTERIDGE, BIRMINGHAM.
Tel: 021-458 4564

128

If you're as **Impressed** as we are with the new **Commodore 128**, you'll be even more **Impressed** with the new software that's appearing for it now! Already we've got **SuperBase 128** (~~£99.95~~ £85.00!), **SuperScript 128** (~~£79.95~~ £69.95!), and the amazing **VizaWrite Classic 128** (~~£99.95~~ £89.00!)... so watch this space!



Although we are by far **the largest UK retailer** of SuperBase, VizaStar and VizaWrite for the **Commodore 64**, for over **five years** we have supported the best software for all the **Commodore** business machines! So if you have a **CBM/PET 3000, 4000, 8000, 700 or PC**, then don't forget us... we haven't forgotten you! Why not phone or write for our **CBM/PET catalogue?**

Superscript 64

All the features you'll ever need for **professional word processing**... and then more! Unrivalled **240-column** text handling, **variable width** screen, **word-wrap, cut-and-paste, four-way** scrolling, document linking for **unlimited capacity**, global **search-and-replace, help screen**, full memory **calculator**, row and column **arithmetic, menu** or **keyed** commands with **command strings, background printing**, complete **letter quality** print control, **spelling checker** with both UK and US **disk dictionaries**, comprehensive **four part** user manual...

ALL FOR ONLY ~~£69.95~~ £61.95!

1st BOOKS

The Anatomy of A Commodore 64	£8.95
The Anatomy of the 1541 Disk Drive	£8.95
Your '64 Cassette Book	£8.95
Tricks and Tips for your '64	£8.95
Machine Language Book for the '64	£8.95
'64 Advanced Machine Language	£8.95
Peeks and Pokes for the '64	£7.95
Commodore 64 Idea Book	£8.95
Graphics Book for your '64	£8.95

Super TYPE

Are you a **two-finger** typist? Then why not let **SuperType** help you learn to **touch-type** the fast and easy way... with your **computer** as the teacher! All the family can learn this useful skill, at their **own pace**... and profit from the very latest computer-aided **training** and **feedback** techniques! For only ~~£19.95~~ £17.95 on tape, or

ON DISK ~~£25.95~~ £22.95!

Superbase 64

Transform your **Commodore 64** into a full featured **professional** database system, with up to **1000** characters per record on up to **four screens**... and up to **128 items** per record, definable as **key, text, numeric, result** or **date**... in files of up to 16 million characters! **SuperBase 64** even has **calculator** and **calendar** functions, **easy input** from word processor or data files, both **menu-driven** and **program** control, **sorting** and **searching**, fully definable **report** and **screen** formats... **Superbase 64** is essential if you want the **most** from your 64! Supplied on 1541 disk with excellent tutorial and reference manual, plus **audio learning** tape...

NOW ONLY ~~£79.95~~ £69.95!

Jetpack

Your programs will really **take off** with **JetPack 64!** 100% compatible with CBM Basic, **Jetpack** compiles Basic programs into **machine code**, running up to **25 times** faster... using **up to half** as much memory! And it even accepts **Basic extensions** and **machine code** routines as well!

OUR PRICE (DISK) ~~£39.95~~ £29.95!

1st ASSEMBLER MONITOR 64

A powerful **2-pass assembler** allowing free-form input, with **31K free** for source code, assembling to **disk** or **memory**... plus loadable **symbol tables**, redefinable symbols, twenty **pseudo-ops, macros, conditional assembly** and unique **16-bit expression evaluator**. The monitor has **15 commands** including **bank switching!**

(ON DISK) ONLY £19.99!

VIZASTAR 64

Spreadsheet, database and **graphics** too... all integrated into one amazing package, at an amazingly **affordable** price! The very latest design techniques combine the ultimate in **ease-of-use** with all the sophistication of a fully **Integrated** product... VizaStar's advanced features include **high speed** maths functions, large **1000 by 64** worksheet, programmability, windows, cell protection, search and sort, text editing, wordprocessor compatibility, simultaneous graphics... integrated with a **1000-character** per record **database**, up to **250 characters** per field... the **only** product of its kind for the **Commodore 64!** Suits most printers, supplied with excellent handbook. (XL8 version now available with a **40% bigger** spreadsheet plus **business graphics**)

OUR PRICE (XL4) ~~£99.95~~ £89.00! (XL8) ~~£129.95~~ £115.95!

VIZAWRITE 64

A **high-performance, low-cost** wordprocessor, with full **on-screen** formatting, that takes full advantage of the **colour, graphics** and **memory** capacity of the **Commodore 64**... that works with both **disk** and **tape**... and supports virtually **any** printer! With a concise and easy-to-follow user manual, **VizaWrite** is the **ultimate** personal computer word processor! Available on **cartridge** for ~~£89.95~~ £78.95, on **disk** for ~~£79.95~~ £67.95, or with **VizaSpell**, a **30,000 word** disk dictionary and spelling checker, for ~~£99.95~~ £85.00!

WANT IT TOMORROW??? CALL US TODAY!!! ON 01-546-7256

SIMON'S BASIC	~~£50.00~~	£35.95
MASTER 64	~~£69.00~~	£39.95
JETPACK (tape)		£14.95
SUPERBASE STARTER	~~£39.95~~	£35.95
EasyScript to SuperScript upgrade - send disk and manual cover		£39.95

FIRST WORD wordprocessor	£35.99
FIRST BASE database	£35.99
POWER PLAN spreadsheet	£44.99
BASIC 64 compiler	£34.99
PASCAL 64 compiler	£34.99
ADA TRAINING COURSE	£49.99

Prices include VAT and POSTAGE and are correct on going to press. Order by post or phone, using cheque, Access, Barclaycard or postal order. Despatch is by same day 1st CLASS post. Product data available on request, or phone for advice if in doubt. [REF A34]

Access VISA

Calco Software

LAKESIDE HOUSE, KINGSTON HILL, SURREY, KT2 7QT. TEL 01-546-7256

Give your data recorder the speed of a disk drive with this program from Nick Hampshire.

FULL SPEED AHEAD

THE MOST TEDIOUS AND frustrating operation on the C64 is using the cassette deck to LOAD or SAVE a program. The operating system tape routines function OK but are so cumbersome and slow that the thought of buying a disk drive soon becomes a matter of very high priority in every user's mind. However, while waiting for your disk drive, all is not lost. With an ingenious piece of software, it is possible to speed up the tape routines to give a loading speed equal to that of a disk drive.

Virtually all C64 software currently being marketed uses some form of fast loader. These fast loaders are given names like: Turbo (this was the first fast loader available), Pavload, Flash Load, etc. The origin of these fast loader routines is rather obscure since many of the software houses use the same loader routines. In this section we give the source code for two fast loaders and their associated SAVE routines. These have been used on several software products of Zifra Software Ltd. under the name of *ZITload* and *ZIFTAload*.

A fast loader is a routine which replaces the existing LOAD and allows a program or data to be loaded from tape at about 10 times normal speed thus making a tape as fast as a disk drive. A fast loader simply changes the format of the pulse sequence stored onto the tape in order to allow a far greater density of information storage per inch.

In order to create a fast loader two programs are needed. Firstly, a fast loader program, which is a fairly short machine code routine loaded at the beginning of a LOAD operation and auto run to LOAD the rest of the program and/or data stored in fast loader format. The second program is a routine to SAVE a program in fast loader format, the fast SAVE.

The first major design problem to be overcome is the storage of each bit on the tape. Each bit is stored as a pulse which goes through a high-low transition (see Figure 1). The length of the total pulse decides whether the bit is a one – A long pulse – or a zero – a short pulse. The bit is flagged in the interrupt register on the falling edge of the pulse.

The loader is a machine code program which runs with the interrupts disabled. It sets a timer between the two lengths, and when the timer runs out the interrupt register is checked to see if the pulse came in or not. If the falling edge of the pulse generates an interrupt before the timer runs out then the pulse was a zero, otherwise it was a one. The bits are then rotated into a byte storage until eight bits have been read, thereby loading a full byte.

Before any bytes can be read and stored, the loader must be in sync with the bits on the tape. A string of zero bits with a single one bit at every byte interval achieves this. The routine then tries to align itself by recognising the value of the byte.

An example of a header byte for aligning would be the value 64, hex $40 or, in binary, 01000000. A series of these bytes is written as the header. Only when this byte has been read in and recognised can the actual program be read without risk of alignment errors.

The program is stored in different ways depending on how much is desired. The simplest way of formatting the file is to first SAVE the two byte load address followed by the two byte end address and then the actual file. The final byte following the end of the file is a checksum calculated by the save routine and also during loading. If the two values are the same, the LOAD was successful. The routine for this form of fast loader is given in Program one.

Another type of LOAD, which uses the same saver but is slower, is the interrupt loader. This method has the advantage of LOADing with the screen on and a foreground program running whilst the main program is loaded. Loaders of this type are: Novaload and Micro Load. The difference is that an interrupt is created when a pulse is read by the tape recorder, and the timer is checked to find out whether the pulse was a zero or a one. The whole LOAD is done in the background allowing a foreground program to play music, run a clock, etc. The foreground program must check at regular intervals to see if the loader has flagged for the end of load. The background LOAD in Program two has only a foreground program which is waiting for the end of LOAD flag to be set.

Fast Tape Routines — Making Them Work

Putting the theory into practice to create the fast loader routines is not difficult. The actual timing for the SAVE routine was not calculated from any theoretical formula but just by trial and error. The only guidelines were that the short pulse should be slightly shorter than half the long pulse, as the waveform of the pulse is evened out by the cassette hardware. The timing value used by the loader is just shorter than the time required before the long pulse reaches its falling edge.

There are two program listings for the C64 in this article, one for each of the two types of LOAD. Each program will SAVE a Basic program in its fast format and automatically put the fast loader routine into the filename where it is stored. When loaded, it will automatically start on the warm start vector. The routines are initialised by SYS(49152). A Basic program can be fast saved by using the SAVE command as normal but with a device number of seven, thus:

SAVE "PROGRAM",7

In addition the first fast LOAD also makes use of the secondary address to auto run the program, thus:

SAVE "PROGRAM",7,1

will cause the program to auto run when loaded back. With both routines, when a program has been saved using one of these fast loader SAVE routines it is unnecessary to LOAD anything before the program: it will LOAD directly from the LOAD command.

An example of how fast these routines can be is shown by the following timing table. This was based on the time taken to LOAD a 26.3K byte Basic program.

Method 1	: 1 minute
Disk	: 1 minute 10 seconds
Method 2	: 1 minute 25 seconds
Normal tape	: 8 minutes 40 seconds

It should also be noted that the SAVE routines for the fast tape operation are considerably shorter than the normal tape routines. One wonders why Commodore has not included these types of fast tape routines in the new machines.

By loading these into your C64 you will be able to take some of the tedium and frustration out of using a tape system. In addition it will also make your programs look far more professional.

This article is extracted from one of the *64 Revealed* series of books by Nick Hampshire and published by Collins.

Program Listing 1

```
033C             !  FAST TAPE SAVE FOR THE 64.
033C             !*******************************
033C             !
033C             !THIS ROUTINE WILL SAVE A PROGRAM
033C             !TO TAPE SO THAT WHEN LOADED BACK
033C             !IT WILL LOAD WITH THE SCREEN ON.
033C             !
C000             *=$C000
C000 A90B                   LDA #<SAVVEC       !CHANGE SAVE VECTOR
C002 8D3203                 STA $0332          ! TO GO TO NEW
C005 A9C0                   LDA #>SAVVEC       ! SAVE ROUTINE
C007 8D3303                 STA $0333
C00A 60                     RTS
C00B             !
C00B 48      SAVVEC         PHA                !SAVE OFF .A
C00C A5BA                   LDA $BA            !GET DEVICE #
C00E C907                   CMP #$07           !NUMBER 7?
C010 F004                   BEQ TSAVE          !YES
C012 68                     PLA
C013 4CEDF5                 JMP $F5ED          !DO NORMAL SAVE
C016             !
C016 A00E    TSAVE          LDY #$0E
C018 A920                   LDA #$20
C01A 9999C1  LOOP1          STA FLNAME+1,Y     !BLANK FILENAME
C01D 88                     DEY
C01E D0FA                   BNE LOOP1
C020 A4B7                   LDY $B7            !GET FILENAME LENGTH
C022 C00F                   CPY #$0F           !GREATER THAN 14?
C024 9002                   BCC LOOP2          !NO
C026 A00E                   LDY #$0E           !ONLY 1ST 14 CHARS
C028 88      LOOP2          DEY
C029 3008                   BMI TSAVE1
C02B B1BB                   LDA ($BB),Y        !GET FILENAME
C02D 999AC1                 STA FLNAME+2,Y     !STORE IT
C030 4C28C0                 JMP LOOP2          !DO NEXT CHAR
C033             !
C033 A058    TSAVE1         LDY #$58
C035 B93FC1  TSAVE2         LDA LOADER,Y       !GET LOADER BYTE
C038 99AB02                 STA $02AB,Y        !STORE IT TO SAVE
C03B 88                     DEY
C03C 10F7                   BPL TSAVE2         !FOR ALL BYTES
C03E A901                   LDA #$01
C040 AA                     TAX
C041 A8                     TAY
C042 20BAFF                 JSR $FFBA          !SET FILE DETAILS
C045 A9BB                   LDA #$BB           !LENGTH OF FILENAME
C047 A298                   LDX #<FLNAME       !FILENAME LO
C049 A0C1                   LDY #>FLNAME       !FILENAME HI
C04B 20BDFF                 JSR $FFBD          !SET NAME DETAILS
C04E A900                   LDA #$00
C050 859D                   STA $9D
C052 A902                   LDA #$02
C054 85FC                   STA $FC            !SAVE START HI
C056 A9AB                   LDA #$AB
C058 85FB                   STA $FB            !SAVE START LO
C05A A9FB                   LDA #$FB           !POINT TO START
C05C A204                   LDX #$04           !SAVE END LO
C05E A003                   LDY #$03           !SAVE END HI
C060 20D8FF                 JSR $FFD8          !SAVE LOADER FILE
C063 A983                   LDA #$83           !RESET WARM START
C065 8D0203                 STA $0302          ! VECTOR
C068 A9A4                   LDA #$A4
C06A 8D0303                 STA $0303
C06D A9FF                   LDA #$FF
C06F 859D                   STA $9D
C071 A200                   LDX #$00
C073 A000                   LDY #$00
C075             !
C075             !THE FAST SAVE ROUTINE
C075             !STARTS HERE.
C075             !
C075 20C6C0  SAVEIT         JSR WRTHDR         !WRITE ALIGNMENT BYTES
C078 A52B                   LDA $2B            !GET START LO
C07A 48                     PHA
C07B 20F5C0                 JSR WRTBYT         !WRITE IT
C07E A52C                   LDA $2C            !GET START HI
C080 48                     PHA
C081 20F5C0                 JSR WRTBYT         !WRITE IT
C084 A52D                   LDA $2D            !GET END LO
C086 20F5C0                 JSR WRTBYT         !WRITE IT
C089 A52E                   LDA $2E            !GET END HI
C08B 20F5C0                 JSR WRTBYT         !WRITE IT
C08E 84FB                   STY $FB            !ZERO CHECKSUM
C090 A42B                   LDY $2B            !GET PAGE OFFSET
C092 A900                   LDA #$00
C094 852B                   STA $2B            !ZERO LO BYTE
C096 B12B    TSAVLOOP       LDA ($2B),Y        !GET A BYTE
C098 20F5C0                 JSR WRTBYT         !WRITE IT
C09B C8                     INY
C09C D002                   BNE TSAVE3
C09E E62C                   INC $2C
C0A0 C42D    TSAVE3         CPY $2D            !CHECK END OF SAVE
C0A2 A52C                   LDA $2C
C0A4 E52E                   SBC $2E
C0A6 90EE                   BCC TSAVLOOP       !NOT YET
C0A8 A5FB                   LDA $FB            !GET CHECKSUM
C0AA 20F5C0                 JSR WRTBYT         !WRITE IT
C0AD 2009C1                 JSR WRTBIT         !CLOSE OFF LAST BIT
C0B0 A91B                   LDA #$1B
C0B2 8D11D0                 STA $D011          !UNBLANK SCREEN
C0B5 A937                   LDA #$37
C0B7 8501                   STA $01            !STOP TAPE
C0B9 58                     CLI                !RESTART  IRQ
C0BA 68                     PLA                !GET START HI
C0BB 852C                   STA $2C            !STORE IT
C0BD 68                     PLA                !GET START LO
C0BE 852B                   STA $2B            !STORE IT
C0C0 2084FF                 JSR $FF84          !RESET I/O
C0C3 4C74A4                 JMP $A474          !EXIT TO 'READY.'
C0C6             !
C0C6 A906    WRTHDR         LDA #$06           !BASIC ROM OUT &
C0C8 8501                   STA $01            ! START TAPE
C0CA A90B                   LDA #$0B
C0CC 8D11D0                 STA $D011          !BLANK SCREEN
C0CF CA      HEADR1         DEX                !PAUSE FOR TAPE
C0D0 D0FD                   BNE HEADR1         ! TO GET TO FULL
C0D2 88                     DEY                ! SPEED
C0D3 D0FA                   BNE HEADR1
C0D5 78                     SEI                !DISABLE IRQ
C0D6 A9A0                   LDA #$A0           !INITIAL TIMER
C0D8 8D04DD                 STA $DD04          ! VALUE FOR DELAY
C0DB A900                   LDA #$00
C0DD 8D05DD                 STA $DD05
C0E0 A919                   LDA #$19
C0E2 8D0EDD                 STA $DD0E          !START TIMER
C0E5 A040                   LDY #$40
C0E7 A940    HEADR2         LDA #$40           !01000000 FOR
C0E9 20F5C0                 JSR WRTBYT         !ALIGNMENT
C0EC 88                     DEY
C0ED D0F8                   BNE HEADR2         !WRITE 64 OF THEM
C0EF A95A                   LDA #$5A           !CHECK ALIGNMENT
C0F1 20F5C0                 JSR WRTBYT         !WRITE IT
C0F4 60                     RTS
C0F5             !
C0F5 85BD    WRTBYT         STA $BD            !STORE BYTE
C0F7 45FB                   EOR $FB            !CHECKSUM
C0F9 85FB                   STA $FB
C0FB A908                   LDA #$08           !LOOP FOR 8 BITS
C0FD 85A3                   STA $A3
C0FF 26BD    WBYTE1         ROL $BD            !BIT INTO CARRY
C101 2009C1                 JSR WRTBIT         !WRITE THE BIT
C104 C6A3                   DEC $A3
C106 D0F7                   BNE WBYTE1         !DO NEXT BIT
C108 60                     RTS
C109             !
C109 A270    WRTBIT         LDX #$70           !ASSUME ZERO BIT
C10B 9002                   BCC WBIT1          !CORRECT ASSUMPTION
C10D A2FF                   LDX #$FF           !SET FOR ONE BIT
C10F 8E04DD  WBIT1          STX $DD04          !SET TIMER
C112 A900                   LDA #$00
C114 8D05DD                 STA $DD05
C117 A901                   LDA #$01           !WAIT FOR TIMER
C119 2C0DDD  WBIT2          BIT $DD0D
C11C F0FB                   BEQ WBIT2
C11E A501                   LDA $01            !TOGGLE WRITE BIT
C120 4908                   EOR #$08           ! IN 6510 REGISTER
C122 8501                   STA $01
C124 EE20D0                 INC $D020          !SHOW IT IS WORKING
C127 A919                   LDA #$19
C129 8D0EDD                 STA $DD0E          !START TIMER
C12C A901                   LDA #$01           !WAIT FOR TIMER
C12E 2C0DDD  WBIT3          BIT $DD0D
C131 F0FB                   BEQ WBIT3
C133 A501                   LDA $01            !TOGGLE WRITE BIT
C135 4908                   EOR #$08           ! IN 6510 REGISTER
C137 8501                   STA $01
C139 A919                   LDA #$19
C13B 8D0EDD                 STA $DD0E          !START TIMER
C13E 60                     RTS
C13F             !
C13F             !THE LOADER STARTS HERE
C13F             !
C13F A005    LOADER         LDY #$05
C141 A920                   LDA #$20           !BLANK OUT 'READY.'
C143 995004  BLLOOP         STA $0450,Y
C146 88                     DEY
C147 10FA                   BPL BLLOOP
C149 78                     SEI                !DISABLE IRQ
C14A A905                   LDA #$05
C14C 8501                   STA $01            !START TAPE
C14E A91F                   LDA #$1F           !DISABLE KEYBOARD
C150 8D0DDD                 STA $DD0D
C153 8D0DDC                 STA $DC0D
C156 AD0DDD                 LDA $DD0D
C159 AD0DDC                 LDA $DC0D
C15C A968                   LDA #$68           !SET TIMER
C15E 8D04DC                 STA $DC04
C161 A903                   LDA #$03
C163 8D05DC                 STA $DC05
C166 A990                   LDA #$90           !ENABLE TAPE IRQ
C168 8D0DDC                 STA $DC0D
C16B A951                   LDA #$51           !SET IRQ VECTOR
C16D 8DFEFF                 STA $FFFE          ! TO POINT TO
C170 A903                   LDA #$03           ! LOAD ROUTINE
C172 8DFFFF                 STA $FFFF
C175 A977                   LDA #$77           !SET NMI VECTOR
C177 8DFAFF                 STA $FFFA          ! TO POINT TO
C17A A903                   LDA #$03           ! AN RTI
C17C 8DFBFF                 STA $FFFB
C17F A900                   LDA #$00           !CLEAR LOADED FLAG
C181 8502                   STA $02
C183 58                     CLI
C184 4CDD03                 JMP $03DD          !WAIT FOR END OF LOAD
C187             !
C187             !*=$02F3
C187             !
C187 A983                   LDA #$83           !RESET WARM START
C189 8D0203                 STA $0302          ! VECTOR
C18C A9A4                   LDA #$A4
C18E 8D0303                 STA $0303
C191 4C84FF                 JMP $FF84          !RESET I/O
C194 8BE3AB                 WOR $E38B,$02AB
C198             !
C198 932A20  FLNAME         TXT "J"
C1A8             !14 SPACES
C1A8 48                     PHA                !IRQ ENTRY POINT
C1A9 98                     TYA
C1AA 48                     PHA
C1AB AD05DC                 LDA $DC05          !GET TIMER HI BYTE
C1AE A019                   LDY #$19           !RESTART TIMER
C1B0 8C0EDC                 STY $DC0E
C1B3 4902                   EOR #$02           !FLIP BIT 1
C1B5 4A                     LSR A              ! AND SHIFT TO
C1B6 4A                     LSR A              ! CARRY
C1B7 26A9                   ROL $A9            !MOVE BIT INTO
C1B9 A5A9                   LDA $A9            ! BYTE READ
C1BB 9002                   BCC BITGOT         !WHEN BYTE READ
C1BD B00D                   BCS EXIT           !NOT COMPLETE BYTE
C1BF C940    BITGOT         CMP #$40           !ALIGNMENT?
C1C1 D009                   BNE EXIT           !NO
C1C3 A916                   LDA #$16           !SET NEW ADDRESS
C1C5 8D6503                 STA $0365
C1C8 A9FE    EX1            LDA #$FE           !GET READY FOR
C1CA 85A9                   STA $A9            ! A NEW BYTE
```

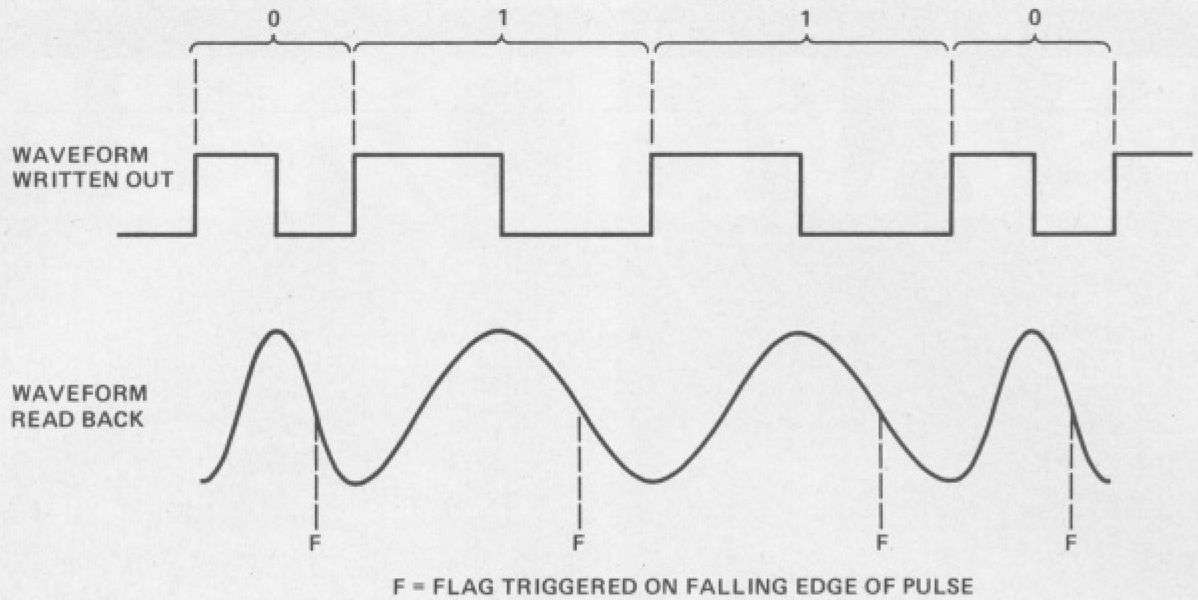


Figure 1. High speed tape waveform

Program Listing 1 (cont.)

```
C1CC AD0DDC EXIT     LDA $DC0D          !CLEAR IRQ
C1CF 68              PLA
C1D0 A8              TAY
C1D1 68              PLA
C1D2 40              RTI                !EXIT IRQ
C1D3          !
C1D3 C940            CMP #$40           !MORE ALIGNMENT?
C1D5 F0F1            BEQ EX1            !YES
C1D7 C95A            CMP #$5A           !FINAL CHECK?
C1D9 F007            BEQ BITGT2         !YES
C1DB A902            LDA #$02           !GO BACK TO
C1DD 8D6503          STA $0365          ! ALIGNMENT ROUTINE
C1E0 D0E6            BNE EX1
C1E2 A930   BITGT2   LDA #$30           !SET NEW ADDRESS
C1E4 8D6503          STA $0365          ! TO READ IN LOAD
C1E7 A900            LDA #$00           ! ADDRESSES. CLEAR
C1E9 85C1            STA $C1            ! CHECKSUM
C1EB F0DB            BEQ EX1
C1ED          !
C1ED 85FB            STA $FB            !STORE LOAD ADDRESS
C1EF EE9703          INC $0397          !INCREASE STORE
C1F2 AD9703          LDA $0397          ! UNTIL 4 BYTES
C1F5 C9FF            CMP #$FF
C1F7 D0CF            BNE EX1            !NOT YET
C1F9 A943            LDA #$43           !STORE NEW ADDRESS
C1FB 8D6503          STA $0365          !FOR READING FILE
C1FE D0C8            BNE EX1
C200          !
C200 A000            LDY #$00
C202 91FB            STA ($FB),Y        !STORE A BYTE
C204 45C1            EOR $C1            !CALCULATE CHECKSUM
C206 85C1            STA $C1
C208 EE00D8          INC $D800          !SHOW IT IS WORKING
C20B E6FB            INC $FB            !INCREASE ADDRESS
C20D D002            BNE BITGT5
C20F E6FC            INC $FC
C211 A5FB   BITGT5   LDA $FB            !END OF LOAD?
C213 C5FD            CMP $FD
C215 A5FC            LDA $FC
C217 E5FE            SBC $FE
C219 90AD            BCC EX1            !NOT YET
C21B A965            LDA #$65           !NEW ADDRESS FOR
C21D 8D6503          STA $0365          !CHECKSUM
C220 D0A6            BNE EX1
C222          !
C222 85C2            STA $C2            !STORE CHECKSUM
C224 A9FF            LDA #$FF           !FLAG END OF LOAD
C226 8502            STA $02
C228 A902            LDA #$02           !RESET BRANCH TO
C22A 8D6503          STA $0365          ! ALIGNMENT
C22D A9FB            LDA #$FB
C22F 8D9703          STA $0397
C232 D094            BNE EX1
C234          !
C234          !*=$03DD
C234          !
C234 A502   PAUSE    LDA $02            !WAIT FOR FILE
C236 F0FC            BEQ PAUSE          ! TO BACKGROUND LOAD
C238 A900            LDA #$00
C23A 8502            STA $02
C23C A907            LDA #$07
C23E 8501            STA $01            !RESET KERNAL ROM
C240 20F302          JSR $02F3


C243 2063A6          JSR $A663          !CLR
C246 A5C1            LDA $C1            !COMPARE CHECKSUMS
C248 C5C2            CMP $C2
C24A D003            BNE LODERR         !DIFFERENT..ERROR
C24C 4C74A4          JMP $A474          !GO TO 'READY.'
C24F          !
C24F A21D   LODERR   LDX #$1D           !'?LOAD  ERROR'
C251 4C37A4          JMP $A437          !SEND ERROR
```

Program Listing 2

```
033C          !  FAST TAPE SAVE FOR THE 64.
033C          !********************************
033C          !
033C          !THIS ROUTINE WILL SAVE A PROGRAM
033C          !TO TAPE SO THAT WHEN LOADED BACK
033C          !IT WILL LOAD FASTER THAN THE
033C          !1541 DISK DRIVE.
033C          !
033C          !AN OPTION FOR AUTO-RUN IS
033C          !INCLUDED.
033C          !
C000          *=$C000
C000 A90B            LDA #<SAVVEC       !CHANGE SAVE VECTOR
C002 8D3203          STA $0332          ! TO GO TO NEW
C005 A9C0            LDA #>SAVVEC       ! SAVE ROUTINE
C007 8D3303          STA $0333
C00A 60              RTS
C00B          !
C00B 48     SAVVEC   PHA                !SAVE OFF .A
C00C A5BA            LDA $BA            !GET DEVICE #
C00E C907            CMP #$07           !NUMBER 7?
C010 F004            BEQ TSAVE          !YES
C012 68              PLA
C013 4CEDF5          JMP $F5ED          !DO NORMAL SAVE
C016          !
C016 A5B9   TSAVE    LDA $B9            !GET SEC. ADDR.
C018 8D29C2          STA RUNFLG         !FLAG FOR AUTO-RUN
C01B A00F            LDY #$0F
C01D A920            LDA #$20
C01F 998CC1 LOOP1    STA FLNAME,Y       !BLANK FILENAME
C022 88              DEY
C023 10FA            BPL LOOP1
C025 A4B7            LDY $B7            !GET FILENAME LENGTH
C027 C011            CPY #$11           !GREATER THAN 16?
C029 9002            BCC LOOP2          !NO
C02B A010            LDY #$10           !ONLY 1ST 16 CHARS
C02D 88     LOOP2    DEY
C02E 3008            BMI TSAVE1
C030 B1BB            LDA ($BB),Y        !GET FILENAME
C032 998CC1          STA FLNAME,Y       !STORE IT
C035 4C2DC0          JMP LOOP2          !DO NEXT CHAR
C038          !
C038 A047   TSAVE1   LDY #$47
C03A B944C1 TSAVE2   LDA LOADER,Y       !GET LOADER BYTE
C03D 99BC02          STA $02BC,Y        !STORE IT TO SAVE
C040 88              DEY
C041 10F7            BPL TSAVE2         !FOR ALL BYTES
C043 A901            LDA #$01
C045 AA              TAX
C046 A8              TAY
C047 20BAFF          JSR $FFBA          !SET FILE DETAILS
C04A A99E            LDA #$9E           !LENGTH OF FILENAME
C04C A28C            LDX #<FLNAME       !FILENAME LO
C04E A0C1            LDY #>FLNAME       !FILENAME HI
C050 20BDFF          JSR $FFBD          !SET NAME DETAILS
C053 A900            LDA #$00           !STOP NAME FROM BEING
C055 859D            STA $9D            !PRINTED ON SAVE
C057 A902            LDA #$02
C059 85FC            STA $FC            !SAVE START HI
C05B A9BC            LDA #$BC
C05D 85FB            STA $FB            !SAVE START LO
C05F A9FB            LDA #$FB           !POINT TO START
C061 A204            LDX #$04           !SAVE END LO
C063 A003            LDY #$03           !SAVE END HI
C065 20D8FF          JSR $FFD8          !SAVE LOADER FILE
C068 A903            LDA #$03
C06A 8D0203          STA $0302          !RESET WARM START
C06D A9A4            LDA #$A4           ! VECTOR
C06F 8D0303          STA $0303
C072 A901            LDA #$01
C074 859D            STA $9D
C076 A000            LDY #$00
```

Program Listing 2 (cont.)

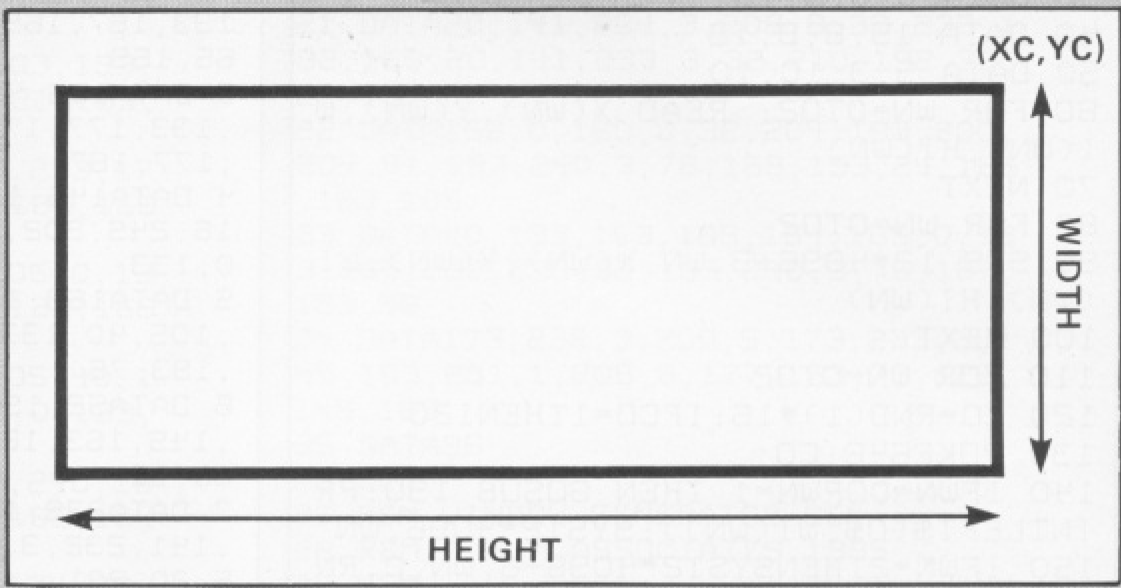
```
C078 A200           LDX #$00
C07A           !
C07A           !THE FAST SAVE ROUTINE
C07A           !STARTS HERE.
C07A           !
C07A 20CBC0         JSR WRTHDR  20CB80  !WRITE ALIGNMENT BYTES
C07D A52B           LDA $2B             !GET START LO
C07F 48             PHA
C080 20FAC0         JSR WRTBYT  20FA80  !WRITE IT
C083 A52C           LDA $2C             !GET START HI
C085 48             PHA
C086 20FAC0         JSR WRTBYT  20FA80  !WRITE IT
C089 A52D           LDA $2D             !GET END LO
C08B 20FAC0         JSR WRTBYT  20FA80  !WRITE IT
C08E A52E           LDA $2E             !GET END HI
C090 20FAC0         JSR WRTBYT  20FA80  !WRITE IT
C093 84FB           STY $FB             !ZERO CHECKSUM
C095 A42B           LDY $2B             !GET PAGE OFFSET
C097 A900           LDA #$00
C099 852B           STA $2B             !ZERO LO BYTE
C09B B12B  TSAVLOOP LDA ($2B),Y         !GET A BYTE
C09D 20FAC0         JSR WRTBYT  20FA80  !WRITE IT
C0A0 C8             INY
C0A1 D002           BNE TSAVE3
C0A3 E62C           INC $2C
C0A5 C42D  TSAVE3   CPY $2D             !CHECK END OF SAVE
C0A7 A52C           LDA $2C
C0A9 E52E           SBC $2E
C0AB 90EE           BCC TSAVLOOP        !NOT YET
C0AD A5FB           LDA $FB             !GET CHECKSUM
C0AF 20FAC0         JSR WRTBYT          !WRITE IT
C0B2 200EC1         JSR WRTBIT          !CLOSE OFF LAST BIT
C0B5 A91B           LDA #$1B
C0B7 8D11D0         STA $D011           !UNBLANK SCREEN
C0BA A937           LDA #$37
C0BC 8501           STA $01             !STOP TAPE
C0BE 58             CLI                 !RESTART  IRQ
C0BF 68             PLA                 !GET START HI
C0C0 852C           STA $2C             !STORE IT
C0C2 68             PLA                 !GET START LO
C0C3 852B           STA $2B             !STORE IT
C0C5 2084FF         JSR $FF84           !RESET I/O
C0C8 4C74A4         JMP $A474           !EXIT TO 'READY.'
C0CB           !
C0CB A906  WRTHDR   LDA #$06            !BASIC ROM OUT &
C0CD 8501           STA $01             ! START TAPE
C0CF A90B           LDA #$0B
C0D1 8D11D0         STA $D011           !BLANK SCREEN
C0D4 CA    HEADR1   DEX                 !PAUSE FOR TAPE
C0D5 D0FD           BNE HEADR1          ! TO GET TO FULL
C0D7 88             DEY                 ! SPEED
C0D8 D0FA           BNE HEADR1
C0DA 78             SEI                 !DISABLE IRQ
C0DB A9A0           LDA #$A0            !INITIAL TIMER
C0DD 8D04DD         STA $DD04           ! VALUE FOR DELAY
C0E0 A900           LDA #$00
C0E2 8D05DD         STA $DD05
C0E5 A919           LDA #$19
C0E7 8D0EDD         STA $DD0E           !START TIMER
C0EA A040           LDY #$40
C0EC A940  HEADR2   LDA #$40            !01000000 FOR
C0EE 20FAC0         JSR WRTBYT          !ALIGNMENT
C0F1 88             DEY
C0F2 D0F8           BNE HEADR2          !WRITE 64 OF THEM
C0F4 A95A           LDA #$5A            !CHECK ALIGNMENT
C0F6 20FAC0         JSR WRTBYT          !WRITE IT
C0F9 60             RTS
C0FA           !
C0FA 85BD  WRTBYT   STA $BD             !STORE BYTE
C0FC 45FB           EOR $FB             !CHECKSUM
C0FE 85FB           STA $FB
C100 A908           LDA #$08            !LOOP FOR 8 BITS
C102 85A3           STA $A3
C104 26BD  WBYTE1   ROL $BD             !BIT INTO CARRY
C106 200EC1         JSR WRTBIT          !WRITE THE BIT
C109 C6A3           DEC $A3
C10B D0F7           BNE WBYTE1          !DO NEXT BIT
C10D 60             RTS
C10E           !
C10E A240  WRTBIT   LDX #$40            !ASSUME ZERO BIT
C110 9002           BCC WBIT1           !CORRECT ASSUMPTION
C112 A290           LDX #$90            !SET FOR ONE BIT
C114 8E04DD WBIT1   STX $DD04           !SET TIMER
C117 A900           LDA #$00
C119 8D05DD         STA $DD05
C11C A901           LDA #$01            !WAIT FOR TIMER
C11E 2C0DDD WBIT2   BIT $DD0D
C121 F0FB           BEQ WBIT2
C123 A501           LDA $01             !TOGGLE WRITE BIT
C125 4908           EOR #$08            ! IN 6510 REGISTER
C127 8501           STA $01
C129 EE20D0         INC $D020           !SHOW IT IS WORKING
C12C A919           LDA #$19
C12E 8D0EDD         STA $DD0E           !START TIMER
C131 A901           LDA #$01            !WAIT FOR TIMER
C133 2C0DDD WBIT3   BIT $DD0D
C136 F0FB           BEQ WBIT3
C138 A501           LDA $01             !TOGGLE WRITE BIT
C13A 4908           EOR #$08            ! IN 6510 REGISTER
C13C 8501           STA $01
C13E A919           LDA #$19            !START TIMER
C140 8D0EDD         STA $DD0E
C143 60             RTS
C144           !
C144           !THE LOADER STARTS HERE
C144           !
C144 AD20D0 LOADER  LDA $D020
C147 85FE           STA $FE             !SAVE BORDER COLOUR
C149 A9A4           LDA #$A4            !RESET WARM START
C14B 8D0303         STA $0303           ! VECTOR
C14E A983           LDA #$83
C150 8D0203         STA $0302
C153 205103         JSR $0351           !FAST LOAD THE FILE
C156 A5FE           LDA $FE
C158 8D20D0         STA $D020           !RESTORE BORDER COLOUR
C15B A937           LDA #$37
C15D 8501           STA $01             !STOP THE TAPE
C15F 58             CLI                 !RESTORE IRQ
C160 A91B           LDA #$1B
C162 8D11D0         STA $D011           !UNBLANK SCREEN
C165 2084FF         JSR $FF84           !RESET I/O
C168 A5FC           LDA $FC             !COMPARE CALCULATED TO
C16A C5FB           CMP $FB             ! LOADED CHECKSUM
C16C D015           BNE LODERR          !DIFFERENT..ERROR
C16E 2063A6         JSR $A663           !CLR
C171 ADDE03         LDA $03DE           !GET AUTO-RUN FLAG
C174 F00A           BEQ EXIT            !NO RUN
C176 208EA6         JSR $A68E           !SET CHARGET POINTER
C179 A900           LDA #$00
C17B 859D           STA $9D             !FLAG RUN MODE
C17D 4CAEA7         JMP $A7AE           !RUN
C180           !
C180 6C0203 EXIT    JMP ($0302)         !WARM START
C183           !
C183 A21D  LODERR   LDX #$1D            !'?LOAD  ERROR'
C185 4C37A4         JMP $A437           !SEND ERROR
C188           !
C188 8BE3           WOR $E38B           !ERROR LINK
C18A BC02           WOR $02BC           !WARM START LINK
C18C           !
C18C 202020 FLNAME  TXT "
C19C           !16 SPACES
C19C           !
C19C           !*=$0351
C19C           !
C19C 208703         JSR $0387           !READ HEADER
C19F 20BA03         JSR $03BA           !READ A BYTE
C1A2 A8             TAY                 !LOAD LO
C1A3 A900           LDA #$00
C1A5 85C1           STA $C1
C1A7 20BA03         JSR $03BA           !READ A BYTE
C1AA 85C2           STA $C2             !LOAD HI
C1AC 20BA03         JSR $03BA           !READ A BYTE
C1AF 852D           STA $2D             !END LO
C1B1 20BA03         JSR $03BA           !READ A BYTE
C1B4 852E           STA $2E             !END HI
C1B6 20BA03 TLOAD1  JSR $03BA           !READ A BYTE
C1B9 91C1           STA ($C1),Y         !STORE IT
C1BB 45FC           EOR $FC             !CALCULATE CHECKSUM
C1BD 85FC           STA $FC
C1BF C8             INY
C1C0 D002           BNE TLOAD2
C1C2 E6C2           INC $C2
C1C4 C42D  TLOAD2   CPY $2D             !CHECK END OF LOAD
C1C6 A5C2           LDA $C2
C1C8 E52E           SBC $2E
C1CA 90EA           BCC TLOAD1          !NOT YET
C1CC 20BA03         JSR $03BA           !READ CHECKSUM
C1CF 85FB           STA $FB
C1D1 60             RTS
C1D2           !
C1D2           !*=$0387
C1D2           !
C1D2 A907           LDA #$07            !START TAPE
C1D4 8501           STA $01
C1D6 A90B           LDA #$0B            !BLANK SCREEN
C1D8 8D11D0         STA $D011           !PAUSE FOR TAPE TO
C1DB CA    RHEAD1   DEX                 ! REACH FULL SPEED
C1DC D0FD           BNE RHEAD1
C1DE 88             DEY
C1DF D0FA           BNE RHEAD1
C1E1 78             SEI                 !DISABLE IRQ
C1E2 84FC           STY $FC             !ZERO CHECKSUM
C1E4 8C05DD         STY $DD05           !SET TIMER HI
C1E7 A9F8           LDA #$F8
C1E9 8D04DD         STA $DD04           !SET TIMER LO
C1EC A200           LDX #$00
C1EE 20C803 RHEAD2  JSR $03C8           !READ A BIT
C1F1 26BD           ROL $BD             !INTO BYTE
C1F3 A5BD           LDA $BD
C1F5 C940           CMP #$40            !ALIGNED?
C1F7 D0F5           BNE RHEAD2          !NOT YET
C1F9 20BA03 RHEAD3  JSR $03BA           !READ A BYTE
C1FC C940           CMP #$40            !IS IT 64?
C1FE F0F9           BEQ RHEAD3          !YES
C200 C95A           CMP #$5A            !ALIGNMENT CHECK?
C202 D0EA           BNE RHEAD2          !NO
C204 60             RTS
C205           !
C205           !*=$03BA
C205           !
C205 A901           LDA #$01
C207 85BD           STA $BD
C209 20C803 GBYTE1  JSR $03C8           !READ A BIT
C20C 26BD           ROL $BD             !INTO BYTE
C20E 90F9           BCC GBYTE1          !COMPLETE BYTE
C210 A5BD           LDA $BD
C212 60             RTS
C213           !
C213           !*=$03C8
C213           !
C213 A910           LDA #$10            !WAIT FOR BIT
C215 2C0DDC GBIT1   BIT $DC0D
C218 F0FB           BEQ GBIT1
C21A AD0DDD         LDA $DD0D           !GET BIT
C21D 48             PHA
C21E A919           LDA #$19            !START TIMER
C220 8D0EDD         STA $DD0E
C223 68             PLA
C224 EE20D0         INC $D020           !SHOW IT IS WORKING
C227 4A             LSR A               !MOVE BIT INTO CARRY
C228 60             RTS
C229           !
C229           !*=$03DE
C229           !
C229 00    RUNFLG   BYT 0
```


A NEW CLASSIC
FROM PAUL WOAKES
AUTHOR OF ENCOUNTER
YOU CHOOSE
THE ACTION IN
MERCENARY
Mercenary – a unique combination of flight simulation, adventure and arcade fun. You choose the action.
There is complete freedom of movement in a truly three-dimensional vector-graphic environment. Graphics of exceptional speed create a very realistic experience.
You'll never play the same game twice. Random elements attempt to seal your fate. Your interaction is crucial.
Mercenary presents an absorbing challenge that you will accept again and again.
MERCENARY
ESCAPE FROM TARG
Fly above the planet Targ in a variety of craft. Investigate the multitude of structures on the planet surface in your quest for essential supplies. Search the mysterious subterranean complex in pursuit of your fortune.
Become embroiled in the continuing conflict between the Palyars and Mechanoids. Accept their paid assignments or work independently to gain wealth.
You only have one true friend – the ever-reliable Benson, an almost human PC, who guides you in your travels.
Pursue your objective as you will : To succeed you have to escape from Targ.
NOVAGEN
INITIAL RELEASE
FOR COMMODORE 64 ATARI 48K 800 XL 130 HE
CASSETTE £9.95
DISK £12.95
NOVAGEN SOFTWARE LTD: SALES 142 ALCESTER ROAD BIRMINGHAM B13 8HS


SCHIZOFRENIA
or how to be in two minds about paranoia
£7.99
SPECTRUM 48K
COMMODORE 64
Alphonse T. Nurd needs help. His alter-ego has discovered the meaning of independence. If you can discover the secrets of sub-atomic particle separators and clean lifts, this is the game for you. The puzzle of a lifetime … or two.
QUICKSILVA
The most powerful games in the Universe!
Available from all good software shops.
In case of difficulty write to:
QUICKSILVA Ltd. Liberty House, 222 Regent Street, London W1R 7DB tel: 01 439 0666
Argus Press Software Group

Let some fresh air into your graphics with this look through windows by Allen Webb.

TOP DRAW



GREETINGS FELLOW 64 OWNERS, HERE are some more graphics wheezes to add that bit of power to your elbows. First, however, it's grovelling time. Remember part one of this series? You may have noticed a POKE to location 1006 in the demonstration. Much to my embarass-ment, I didn't tell you what it does. It holds the number of shifts executed when you call the roll routines. A value of one will roll the design one pixel, a value of two rolls it two pixels, etc. Values greater than three aren't too helpful, but the use of this register will make your efforts that little bit more interesting.

This month I want to deal with graphics windows. A window is a definable area of the screen which can be manipulated independently of the rest of the screen. The real value of windows is in the generation of menus and text games such as adventures.

As usual, the software is given as a Basic loader in listing one. The code sits at $C000 so it will unfortunately clash with the character routines – abject apologies but such is life.

The first problem to resolve is how to define a window. Consider Figure 1. The rectangle represents a window of width W characters and height H units somewhere on the screen. The top left hand corner is fixed by the co-ordinates XC and YC. These four values are all that is necessary to specify the window. The machine code allows you to reserve and access up to 10 windows – numbered 0 to 9. Each window is defined in four tables in terms of co-ordinates of the top left hand corner and the height and width. The command for specifying a window has the syntax:

```
SYS 49155,WN,XC,YC,W,H
```

WN is the window number and the other parameters are as specified earlier. It is important that you set up a window before manipulating it. When the program is set up the window tables contain zero and any action on an un-defined window will, at best, corrupt your Basic program so be warned!!

What can you do with the window? The next command has two forms depending on what you wish to do. A flag decides your action and the syntax. The first form will fill the window with a specified character CH:

```
SYS 49158,WN,0,CH
```

i.e. the flag has a zero value. A non-zero flag simply reverses the contents of the window:

```
SYS 49158,WN
```

If you use the fill command with a value of CH equal to 32, the window will be cleared.

Finally, you can scroll the contents of the screen. To maintain compatibility with the normal 64 operation, the scrolling is upwards with the bottom line of the window being filled with blanks. The syntax of this command is:

```
SYS 49152,WN
```

To help you see what these routines will do, I've given two demonstrations. The first shows the manipulation of three windows at once. Since the window scrolls upwards, you must somehow print text at the bottom line. In my experience, the use of cursor control codes and TAB are both inelegant and tiresome. It's far better to use a routine in the 64's Kernal. The following line will do this:

```
POKE 781,Y: POKE 782,X: POKE 783,0: SYS 65520
```

where X and Y are the co-ordinates of the point to which you want to move the cursor. Line 230 in the demo does this.

The second demonstration uses randomly created windows to produce a pattern.

I have included a degree of checking in the routine so that some illegal values (e.g. width 40 or height 25) will be rejected. Not all possibilities are catered for and it's up to you to ensure that silly values are not generated in your program.

The routines should be of most value to adventure freaks since it is simple to create two or three text windows with independent scrolling.

Now, I want to deal with a feature which the BBC, C16 and Spectrum have in common. The ability to use flashing characters. The simple way to implement this is to use the attribute of each character to hold a flash flag. This is not simple to do on the 64 because there is no spare bit in the video matrix and it isn't easy to use the top four bits in the colour matrix. My solution is to let you select a colour to flash. This colour is kept in location 1000. The routine is called every 50th of a second by the IRQ interrupt. The screen is scanned every 25 interrupt calls, so that the flash rate is about twice a second. The routine scans the colour matrix and inverts every character of the specified colour. This routine is given in Listing two and a demonstration in demo three.

Owners with new ROM 64s should take a little care. These machines fill the colour matrix with the current colour each time the screen is cleared. If the current colour is the flash colour, the whole screen will flash. Demo three shows how to use the routine.

To turn off the character flash, simply enter:

```
SYS 52736
```

That's all for this time, see you again next month. Happy hacking!

Demonstration 1

```
0 REM DEMONSTRATION 1
1 REM
10 POKE53281,1
20 O$="******************"
30 DATA 0,0,3,10
40 DATA 15,8,9,12
50 DATA 5,3,10,10
60 FOR WN=OTO2: READ X(WN),Y(WN),W
I(WN),HI(WN)
70 NEXT
80 FOR WN=OTO2
90 SYS 12*4096+3,WN,X(WN),Y(WN),WI
(WN),HI(WN)
100 NEXT
110 FOR WN=OTO2
120 CO=RND(1)*16:IFCO=1THEN120
130 POKE646,CO
140 IFWN=OORWN=1 THEN GOSUB 190:PR
INTLEFT$(O$,WI(WN)):SYS12*4096,WN
150 IFWN=2THENSYS12*4096+6,WN,0,RN
D(1)*256
160 IF RND(1)<.5THEN SYS 12*4096+6
,WN,1
170 NEXT
180 GOTO 110
190 POKE781,Y(WN)+HI(WN)-1 :POKE78
2,X(WN):POKE783,0:SYS65520:RETURN
```

Demonstration 2

```
0 REM DEMONSTRATION 2
1 REM
10 XS=INT(RND(1)*20)+1
20 WI=INT(RND(1)*20)+1
30 YS=INT(RND(1)*12)+1
40 HI=INT(RND(1)*12)+1
50 SYS12*4096+3,1,XS,YS,WI,HI
60 SYS12*4096+6,1,0,RND(1)*128
70 SYS12*4096+6,1,1:GOTO10
```

Demonstration 3

```
10 REM DEMONSTRATION 3
20 REM
30 SYS 52736: REM TURN THEM ON
40 POKE 1000,1 :REM WHITE TO FLASH
50 PRINT"[CLEAR][CYAN]THIS [WHITE]
IS[YELLOW] A [WHITE]DEMONSTRATION[
c 7] OF  [WH
ITE]FLASHING[c 8]    CHARACTERS"
```

Listing 1

```
0 REM LISTING 1
1 DATA76,9,192,76,114,192,76,133,1
93,32,238,192,165,20,141,232,3,32,
210,192
2 DATA32,29,193,24,165,163,105,40,
133,167,165,164,105,0,133,168,24,1
65,169
3 DATA105,40,133,176,165,170,105,0
,133,177,174,92,193,172,91,193,136
,177,167
4 DATA145,163,177,176,145,169,136,
16,245,202,240,28,24,165,163,105,4
0,133
5 DATA163,144,2,230,164,24,165,167
,105,40,133,167,144,2,230,168,32,6
,193,76
6 DATA52,192,172,91,193,136,169,32
,145,163,169,1,145,169,136,16,245,
96,32
7 DATA238,192,165,20,201,10,176,64
,141,232,3,32,238,192,172,232,3,16
5,20,201
8 DATA41,176,49,153,93,193,32,238,
192,172,232,3,165,20,201,25,176,34
,153,103
9 DATA193,32,238,192,172,232,3,165
,20,201,41,176,19,153,113,193,32,2
38,192
```

Listing 2

```
0 REM LISTING 2
1 DATA169,25,141,233,3,120,169,79,
141,20,3,169,206,141,21,3,88,96,16
9,0,133
2 DATA251,169,216,133,252,160,0,17
7,251,41,15,205,232,3,208,17,165,2
51,133
3 DATA253,56,165,252,233,212,133,2
54,177,253,73,128,145,253,230,251,
208,2
4 DATA230,252,165,251,201,232,240,
3,76,28,206,165,252,201,219,240,3,
76,28
5 DATA206,96,206,233,3,208,8,169,2
5,141,233,3,32,18,206,76,49,234,23
8
6 FORI=52736TO52831
7 READX:T=T+X
8 POKE I,X:NEXT
9 IFT<>13738THENPRINT"ERROR IN DAT
A"
10 REM
11 REM FLASHING CHARACTERS ROUTINE
```

```
10 DATA172,232,3,165,20,201,26,176
,4,153,123,193,96,169,195,160,192,
32,30
11 DATA171,96,73,76,76,69,71,65,76
,32,86,65,76,85,69,63,0,172,232,3,
185,93
12 DATA193,141,89,193,185,103,193,
141,90,193,185,113,193,141,91,193,
185,123
13 DATA193,141,92,193,96,32,253,17
4,32,138,173,32,247,183,96,165,169
,24,105
14 DATA40,133,176,165,170,105,0,13
3,177,96,24,165,169,105,40,133,169
,144,2
15 DATA230,170,24,165,176,105,40,1
33,176,144,2,230,177,96,169,0,133,
163,169
16 DATA4,133,164,174,90,193,240,16
,24,165,163,105,40,133,163,165,164
,105,0
17 DATA133,164,202,208,240,174,89,
193,240,14,24,165,163,109,89,193,1
33,163
18 DATA165,164,105,0,133,164,165,1
63,133,169,165,164,24,105,212,133,
170,96
19 DATA10,2,10,10,0,5,0,0,0,0,0,0,
0,0,0,5,0,0,0,0,0,0,0,0,5,5,0,0,0,
0,0,0
20 DATA0,0,5,5,0,0,0,0,0,0,0,0,32,
238,192,165,20,141,232,3,32,238,19
2,165
21 DATA20,141,234,3,208,8,32,238,1
92,165,20,141,233,3,32,210,192,32,
29,193
22 DATA162,0,160,0,32,204,193,200,
204,91,193,240,3,76,169,193,24,165
,163,105
23 DATA40,133,163,165,164,105,0,13
3,164,232,236,92,193,240,3,76,167,
193,96
24 DATA173,234,3,208,5,173,233,3,1
45,163,201,1,208,6,177,163,73,128,
145,163
25 DATA96
26 REM
27 REM WINDOW GENERATOR C64
28 REM A + M ASSOCIATES 1985
29 REM
30 FOR I = 49152 TO 49632
31 READ X: T = T + X
32 POKE I,X
33 NEXT
34 IF T<>55680 THEN PRINT"ERROR IN
 DATA"
```


FAST LOADERS TO DISK — COMMODORE 64 1541 DISK DRIVE — TAPE FAST LOADERS

MEGA-TRANSFER DISK — £16.00

A mighty collection of routines which will efficiently transfer an extremely wide range of fast loading tapes to disk. General purpose routines to transfer Nova, Pav, Burner, Flash, Hyper, CBS loaders and others. A huge menu of individual routines for unusual loaders, including combination slow/fast loaders, the latest sports simulations and top arcade and adventure games. General purpose routine for those difficult U.S. slow loaders which even Diskus 1 can't handle. And transfer routines for programs recorded using R.B.S:, Fastback and Hypersave-64. Includes a program identifier. All transfer is automatic. No user knowledge required. MEGA-TRANSFER DISK is the finest transfer utility available for fast loading programs. Don't settle for less.

MEGA-UTILITY DISK — £11.00

1. ALPHALOAD Disk Turbo Loader. Add this short program to each of your disks and your programs will load at over FOUR times the normal rate. No menu is required and you DON'T have to load Alphaload separately. Example: 200 blocks. Normal speed 130 seconds - with ALPHALOAD just 30 seconds.

2. FAST COPY. A high class single drive whole disk copier. Backup a full disk in a little over 3 minutes. Will handle many commercial disks but not if heavily protected.

3. A multi option FAST disk formatter and a single file transfer routine which will handle program files of any length.

4. DISK TO TAPE. Transfer any single disk program to tape (fast load). No memory conflict. Will handle programs of 200 blocks and more.

DISKUS I — £11.00

The Premier tape to disk transfer utility for your slow load tape collection. Multipart, Headerless, and autorun programs are all catered for. No user knowledge required. Diskus 1 has provision for program types which no other utility can handle. Supplied on disk.

DISCOUNTS!! Any two of MEGA-TRANSFER DISK, DISKUS I or MEGA-UTILITY DISK (single disk) £23.00. ALL THREE just £29.00. Or try our FULL RANGE DISK (every program in this Ad. Plus our high quality Sprite Editor, PRO-SPRITE, and tape backup/header reader at £35.00 (disk only). Megadisk owners. Send Megadisk instruction sheet or Megadisk for £12 credit against any offer in this Ad over £20.

!! TAPE SPECIAL !!

RAPID BACKUP SYSTEM (R.B.S). Convert your slow loading tapes to TURBOLOAD. Multipart and autorun programs are handled with ease. No user knowledge required. R.B.S. will convert more programs than any competing utility.

HYPERSAVE 64. Retaining the flexibility which the programmer requires, Hypersave leaves you in control. Save/load/verify up to 48k continuous Ram at HYPERSPEED.

SPECIAL SUMMER OFFER! BOTH PROGRAMS JUST £7.00 if you mention this Ad. (Programs available individually at £5.00 each)

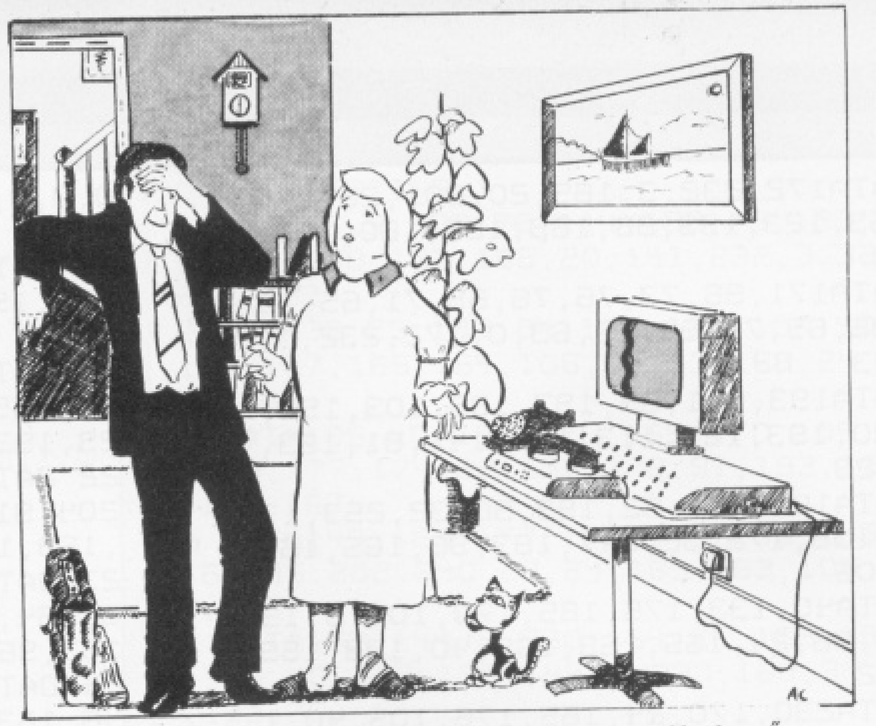
All programs for the Commodore 64.
Send SAE for full details or cheque/P.O. for fast despatch to:

DOSOFT
2 OAKMOOR AVENUE
BLACKPOOL FY2 0EE

U.K. Postage included. Europe please add 75p. Overseas add £1.50 for airmal.

COMPETITION

Want to get your C64 on line? Do you feel the need to communicate with other 64 owners? Well you could win one of 10 Commodore modems that we are giving away in this month's competition.



"Darling, it's been on that phone all day"



"Darling, it's been on that phone all day"

The rules

ENTRIES WILL NOT BE ACCEPTED FROM employees of Argus Specialist Publications, Commodore Business Machines and Alabaster Passmore and Sons. This restriction also applies to employee's families and agents of the companies.

The how to enter section forms part of the rules. The Editor's decision is final and no correspondence will be entered into.

How to enter

There are a few differences between the two pictures. Study them carefully and circle the differences on the picture attached to the coupon. Fill in the coupon clearly and seal the picture and coupon in an envelope. Write clearly the number of differences you found on the back of the envelope.

Post your entry to:
Modem Competition, Your Commodore, No 1 Golden Square, London W1R 3AB. The closing date is first post on Thursday, 31st November 1985. You may enter as many times as you like but each entry must be on an official coupon – not a copy – and sealed in a separate envelope.

Important: Please follow closely the guidelines on entering – incomplete coupons and entries with no numbers on the back cannot be considered. If you are a winner, the coupon will act as a label for your prize so clear writing is essential.

ENTRY FORM

Name ....
Address ....
....
....
....Post Code..........
Age......Occupation ....
Please complete the following tie breaker:
I would use my Commodore modem to ....
....
....
....
....
....



THE COUNTDOWN HAS BEGUN
CRITICAL MASS
From DURELL
Commodore 64
Spectrum
R.R.P. £8.95
DURELL sales dept.,
Castle Lodge, Castle Green, Taunton TA1 4AB
Write now for details of Durell's
Great New Competition also
shirts, Catalogue
Posters

Peter Thomas has been exercising his brain on Ocean's Summer Games II. His verdict was very good. Read on.

FOR THOSE OF YOU, LIKE ME, WHO think that physical exercise is moving the car an extra yard or so from the front door each morning but who are mentally capable of giving Daley Thompson and Steve Cram a run for their money, *Summer Games II* is just perfect.

Owners of the original *Summer Games* – now world leaders in the pole vault, dab hands at the full tuck dive, champions in the 4 × 400 metres relay and challenging Carl Lewis's 100 metre record – beware! If you have thrown in the towel in the freestyle relay and consider the gymnastics and sheet shooting difficult to master, rest assured that you can sit back and relax with Summer Games II – because it's twice as difficult.

You start with a simple hop, step and jump in the triple jump and if you don't fall head first in the sand you can watch an action replay of your successful soar across the sand while the appreciative crowd applauds.

Oarsmen and women among you can tackle the single sculls rowing and pit your skill against your friends, the computer or the clock. The split screen makes this event very exciting and realistic.

After the water you return to dry land for the javelin where a combination of speed, timing and power sees the javelin fly through the air across the screen. Again a good throw gains appreciation from the crowd.

The equestrian event brought back memories of pony trekking on Dartmoor when again I seemed to spend more time on the ground than on the horse. Timing is essential if you don't want an early ducking in the water jump.

Once you've had enough of horsing around you can attempt the high jump. I'm not sure whether there was a bug in my version of the game or whether I'm just not cut out for the high jump, but I only managed to get over the bar once and that was on the lowest level.

If you've got a grudge against someone then you'll love the fencing. Now you've got your chance to flex your foil and battle it out with your opponent. Control is a little difficult but after some practice, and a few beatings, you'll get used to it.

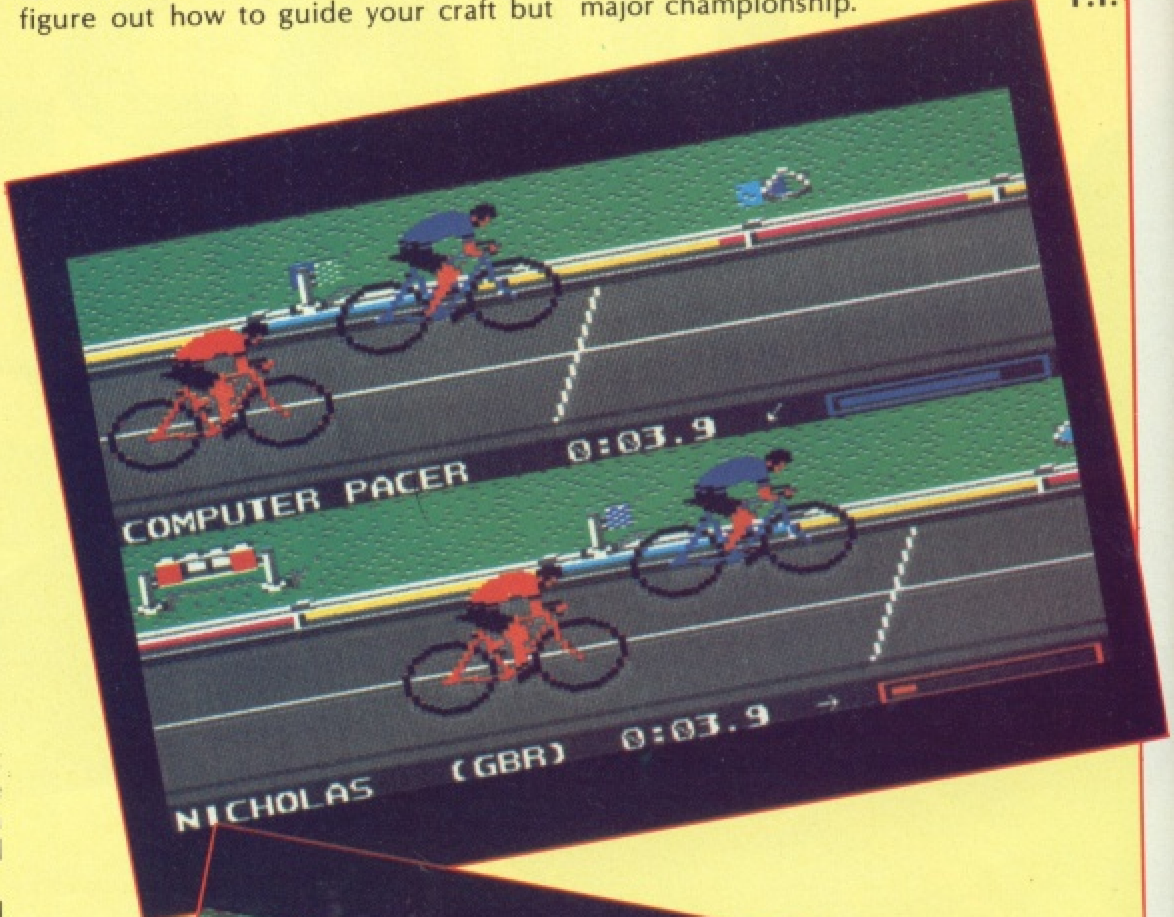
If you find that you have trouble staying on a tricycle then the cycling event is certainly not one for you. Rotating your joystick moves the man's legs on the pedals. If you don't move your joystick at a steady pace then you'll find yourself getting nowhere and see your opponent disappear towards the finishing line.

Kayaking is certainly not an event to be rushed. You must guide your kayak down the white water while passing through a number of gates. Some gates must be passed through forward, others backwards, some you even have to guide your kayak through while going upstream. This is certainly an event which takes a lot of concentration. You'll soon figure out how to guide your craft but getting through the gates is a different matter.

In each event you strive for gold silver or bronze medals all the time attempting new heights of glory to become the world record holder.

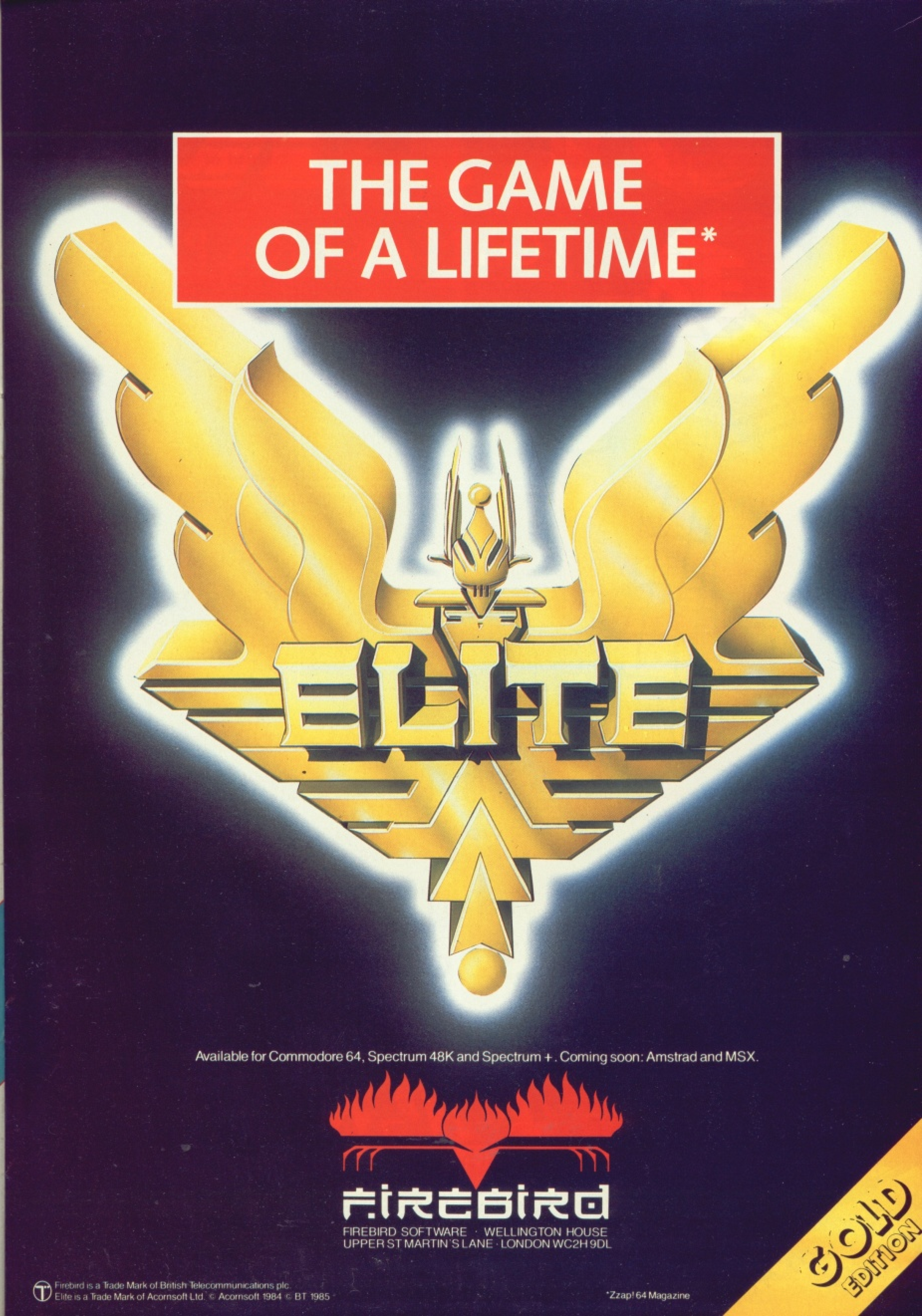
Choose which country to represent out of a possible 18, put on your sportswear and prepare for *Summer Games II*. Alternatively, lock your bedroom door switch on your computer and practice each event for half an hour while going on a strict diet until you are ready for the next major championship.

P.T.





GAME OF THE MONTH

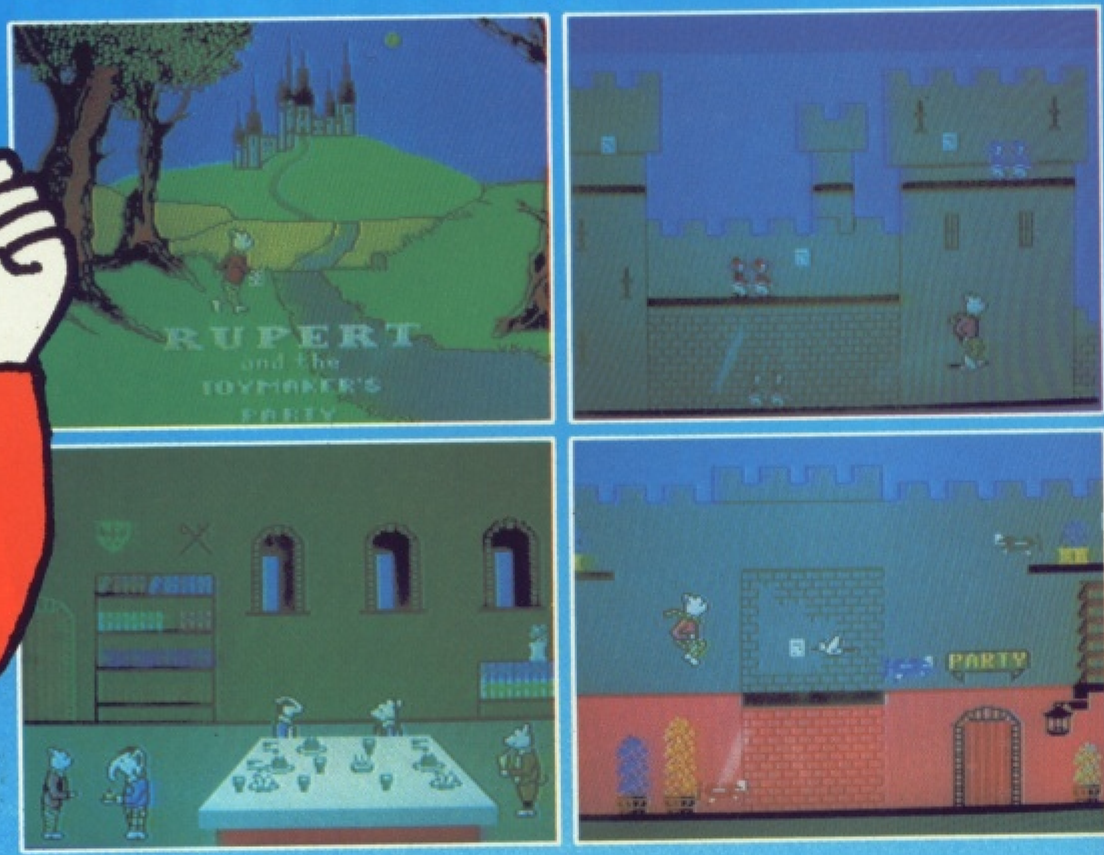

THE GAME
OF A LIFETIME*
ELITE
Available for Commodore 64, Spectrum 48K and Spectrum + . Coming soon: Amstrad and MSX.
FIREBIRD
FIREBIRD SOFTWARE · WELLINGTON HOUSE
UPPER ST MARTIN'S LANE · LONDON WC2H 9DL
GOLD EDITION
Firebird is a Trade Mark of British Telecommunications plc.
Elite is a Trade Mark of Acornsoft Ltd. © Acornsoft 1984 © BT 1985
*Zzap! 64 Magazine

RUPERT

and the

TOYMAKER'S PARTY





Jolly old Rupert can walk, jump, leap, climb stairs – and even fly – but can you get him to the party?

32 challenging screens of animation and fun.

QUICKSILVA

The most powerful games in the Universe

Quicksilva Ltd, Liberty House,
222 Regent Street, London WIR 7DB.
Telephone 01-439 0666

£7.99

Spectrum 48K
CBM64

Rupert Bear © Express Newspapers PLC



Runecaster risks life and limb yet again in order to bring you his column.

AS I AM SURE YOU ALREADY KNOW, adventure games can cost quite a lot of money! Generally speaking the more expensive games are disc based and provide you with many hours of pleasure for your money. The higher middle range of tapes (around £10 plus) seem to be the ones that sometimes make you think.

On the other hand there are a number of bargains around. My local Spectrum Chain shop had a couple of good buys... One would be loosely (very loosely) classed as an arcade adventure – *Mystic Mansion* – and the other, a well presented text adventure, was *Aliard's Tomb*. Respectively, £2.99 and £1.99.

The first, US Gold, hardly comes within the range of this column but is worth looking at, especially if you have young children around. It is a 20 level, find the treasures, avoid the monsters, type of game with a good difficulty grading – Adult, Teenager or Child.

The second, *Aliard's Tomb*, is a neat little adventure that does not appear to have any hideous inconsistencies and at the price is well worth buying. Fairly conventional scenario...dark dungeons, in which you have to find the aforesaid Tomb, solving several ingenious puzzles in the process.

Keep your eyes skinned for these and other 'cheapies', they are often either reissues or are stock from defunct software houses. You will not find *Exodus: Ultima III*, *The Hobbit* or *Eureka* but may well find something to keep you interested.



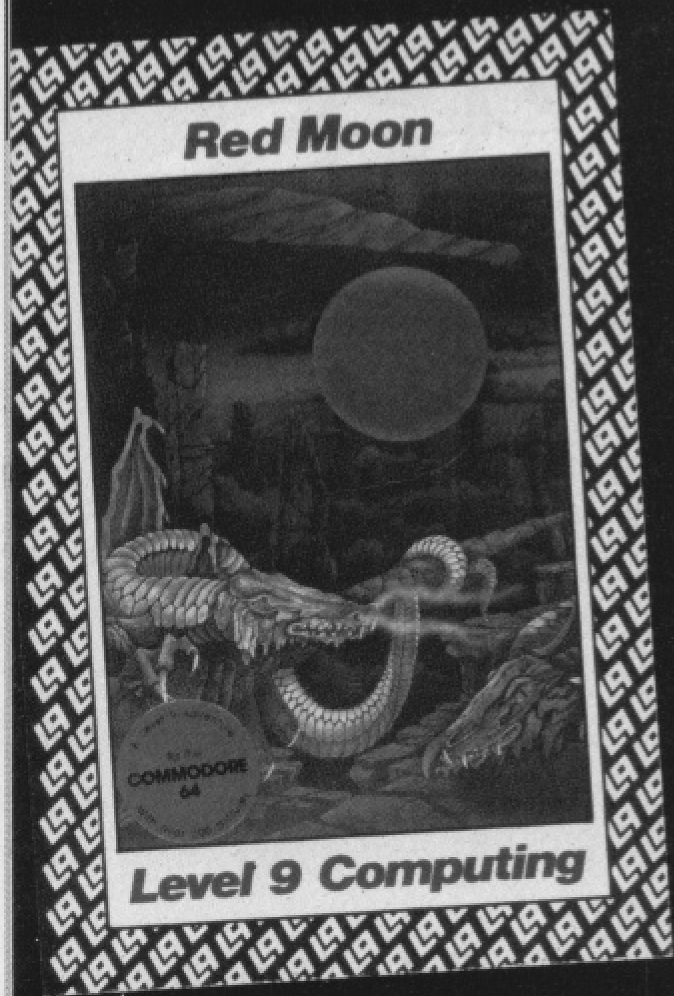


Kitchen spin-off...

Down in the kitchen something stirred...or rather squeaked! It was probably the lady of the house using one of those reusable shopping list boards – you know the type: plastic-faced and supplied with a water soluble ink marker. Make notes throughout the week on what is required – buy them on Saturday and wipe the 'slate' clean on Sunday.

What has this to do with playing adventures? Well, a firm in Wales, Mapit, has produced a plastic reusable Map Maker. It is A3 size (two pages of Your Commodore) and has a 13×11 grid of 14×27mm rectangles printed on it...just the thing for mapping all your trips elsewhere! There is space at the top of the board to make 'vital notes' and the pens supplied have a fine enough point to enable all the vital facts of a location to be recorded in the space provided.

The Map Maker costs £3.49 and there also is a range of suitable pens in various colours at 99p each. A little more expensive than scrap paper but as we approach Christmas...Mapit can be found at: 166 Robert Street, Ynysybwl (perhaps we should have a checksum on that!), Mid Glamorgan CF37 3EA.



Over the moon

Level Nine does it again. *Red Moon* is the latest from Britain's number one adventure software house. As usual the products are available for a number of computers but of interest to us here is the C64 version.

Over 200 pictures can be found as you explore the 'magik' land of the Red Moon. Level Nine seems to improve with age, not only can you turn the pictures off, saving the 'drawing' time, but with this program you can even type in commands whilst the pictures are being created on the screen!

The *Red Moon* is about a land where Magik works, albeit nowhere nearly as strongly as in the past. In the past the moon was crimson red and the source of all the great Magiks. As the moon faded to the grey we know today, so the Magik wanted. A great concourse of Magicians created a substitute – the Red Moon Crystal. This was mounted in the Moon Tower of Baskalos and shone out over the kingdom, maintaining an island of Magik and enlightened Civilization.

But...the great Crystal was stolen. The Red Moon is the story of how a Magician (you!) recovered this fabulous gem and restored hope to the World.

Not only are there hundreds of scene setting pictures but there are also the long and descriptive texts that have always been the hallmark of Level Nine games. Read, and wonder at it all...how do they get it all into our computer's memory? The pictures are good, better than the rather surrealistic landscapes of Return to Eden. They are certainly not high, hi-res works of art but serve the very important purpose of visually triggering recognition of your present location.

As with all Level Nine games (and most other good adventures!) it is important to keep a careful map of your progress. There is a temptation when recognizable pictures are displayed, to skip the mapping procedures...fight this temptation – draw a map!

Unlike many games where an apparently insoluble puzzle bars your progress within a few steps of your entry point, *Red Moon* has plenty of locations for even the absolute novice to explore right from the start. Do not think that this implies a beginners game, even accomplished adventurers will be gnashing their teeth on occasions. For those that get totally lost and confused, Level Nine offers the best hint sheets in the business. These are so laid out that it is not easy to read something you wish you hadn't!

Artefacts found soon after entry will include the inevitable lamp for subsequent exploration in dark and dangerous places. Dangerous? Yes! There are a number of decidedly aggressive creatures just waiting for you to stumble by. To be successful you will have to fight them. Bare hands are not to be recommended, find a useful weapon and better still some armour to protect you. You start your foray into this other world with some 50 'hit points', it is surprising how quickly these can be used up. The use of Magik is permitted, even encouraged. To implement these spells you must enter a command of the type 'CAST ZAP'. Each spell may only be cast successfully if you have the object that is the 'focus' for that particular spell.

'CAST ZAP' is used to magically attack an enemy but you must have the jewelled dagger for it to work. Casting spells also uses up some of your 'hit points', so try not to be too extravagant with them.

SAVE and RESTORE your current position are also counted as spells! In this case no 'focus' is required but it still uses up your 'hit points'. A clever dodge this, as it stops us making dozens of 'safety saves'.

Neither can you cast a spell successfully if you have iron about your person (or at that location). You cannot fault the logic, even if it is a little frustrating at times!

The C64 tape version utilizes a 'rapid loader' and takes just less than three minutes to load this massive program. So, even though disc versions are usually available the gain is fairly small.

Level Nine is to be congratulated on keeping standards at such a high level for such an extended time...good luck and roll on the next one – but please give me a little time to solve this one first!

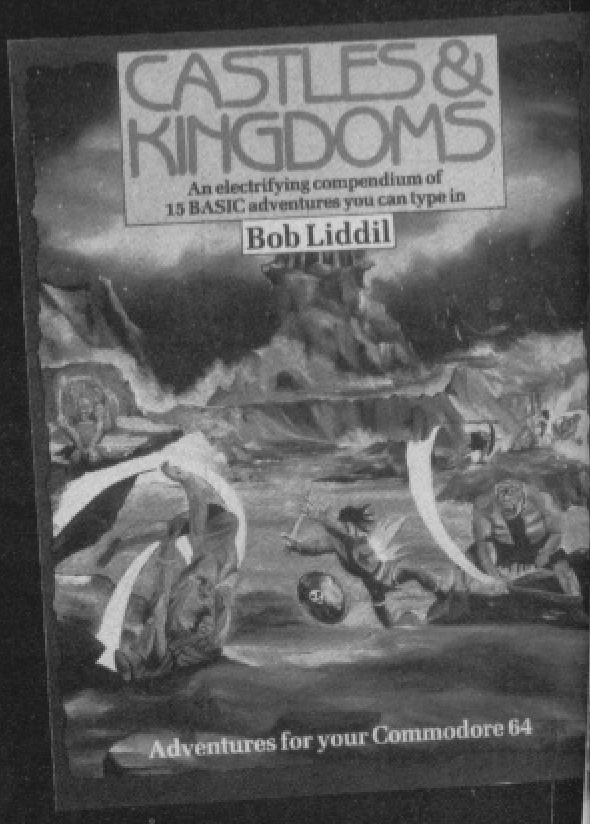
Sore fingers...

Although the range of good software grows as time goes by, there are always some people who like to 'do their own thing'. Obviously if you get **that** good at it we will start reading about your work in this column!

Many others spend enjoyable hours pounding their keyboards typing in programs gleaned from magazines or books. This can often be very, very good programming practise. This is the way to learn how a program works (often the hard but permanent way – by having to debug it!).

It is also often possible to see how to improve on the published listing. The original programmer was probably heavily involved in his program, whereas you will be looking at each section in a fresh light.

There are already several books on the market that will help you write adventure games on the Commodore 64. On the other hand, new ones are always welcome, you never know what new routines you may find! Book is nowadays are rarely cheap but in terms of time spent at your computer are probably much cheaper than the average piece of software. Two offerings have come our



way recently, one American and one British.

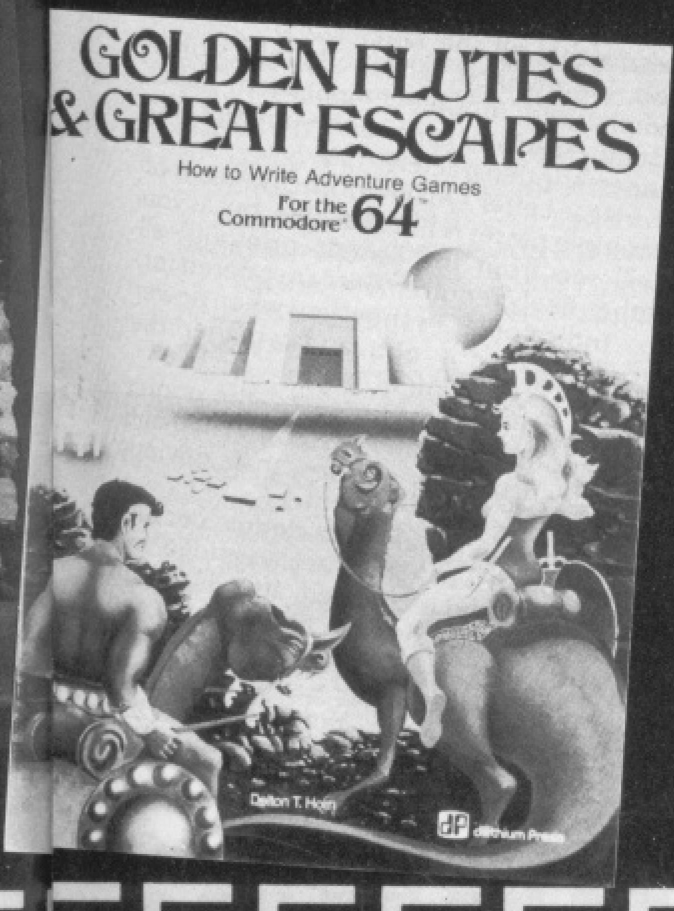
The first is: *Golden Flutes & Great Escapes* by Delton T Horn, published by dilithium Press, ISBN 0-88056-051-7. There are over 200. Four complete listings are given together with hints, tips, flow charts and possible variations to the game play.

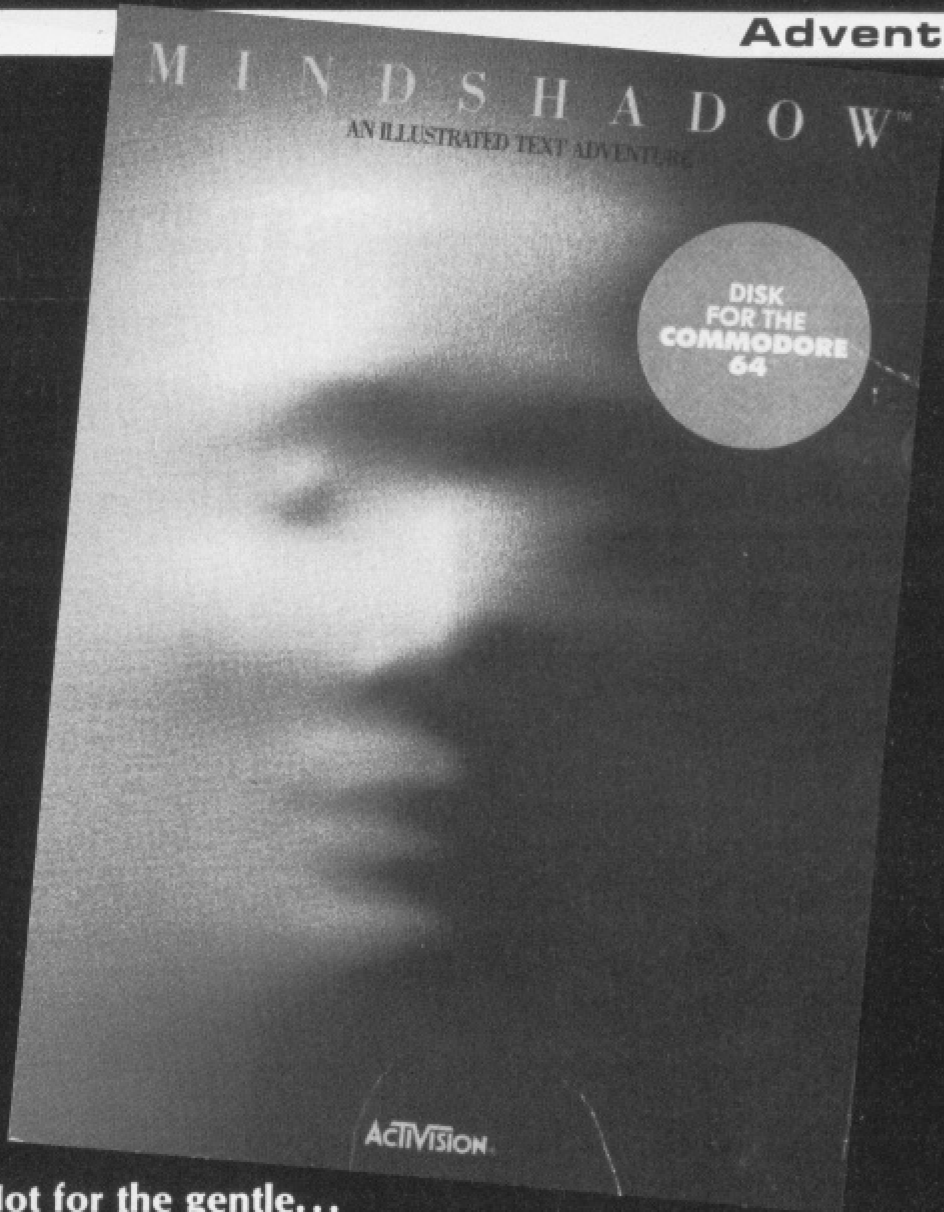
The book is written in clear English (American?) and presents the writing of adventure games in a modular form that should be understood by the reader. The listings are in BASIC and are for text only displays. Although I have not keyed any of them in, they look reasonably interesting if you are just starting out along this path. They should give you plenty to think about. Dilithium Press Publications are distributed in the United Kingdom by: Holt Saunders Ltd, 1, St Anne's Road, Eastbourne BN21 3UN. Holt Saunders has a number of titles dealing with computers and computing, lists of these are obtainable from the above address.

The second book is *Castles and Kingdoms* by Bob Liddil, published by Virgin Books, ISBN 0-86369-094-7 – 186 pages cost £5.99. This book is quite different to the one above as no explanation of each individual program is given.

Listings for 15 (!) different adventures are given together with a page (210×265mm) or so to set the scene for each. Again the programs are in BASIC and are clearly set out for you to type in. The games are presented for you to play rather than learn from but the inevitable debugging is bound to teach you something.

Both books appear to offer good value for money – look out for them and let us know how you get on! Both publications offer copies of the games on cassette for the lazy amongst us!





Not for the gentle...

Do you remember Dallas by US Gold? It appeared in this country last year and was acclaimed by many reviewers at the time. If you have not played it, try to get your computer shop to demonstrate it for you. It is disc based only and has excellent graphics that are called from the disc as you proceed.

What made me think of Dallas is a recent issue from Activision – Mindshadow. It too is disc based and also has good graphics, again called from the disc. The plot is novel too, you awake on a desert island...not knowing who you are!

The aim of the game is to discover your own identity and who left you to perish... As the story unfolds you find yourself travelling around the world in search of the answers. The program is well conceived and presented, the puzzles are good and the use of the function keys well thought out.

The only thing I have against Mindshadow is the underlying need to behave in a somewhat underhand manner to succeed! As a barbarian clobbering trolls or a thief stealing deftly through dungeons – I have no qualms... but striking out with no provocation or stealing from a sleeping tramp...I dunno. That gripe apart this is a good adventure and the use of the function keys make it very easy to play. The most used commands are immediately at your fingertips. SAVE, LOAD, REPEAT, HELP, DROP, GET, QUICKSAVE and QUICKLOAD are all function key commands.

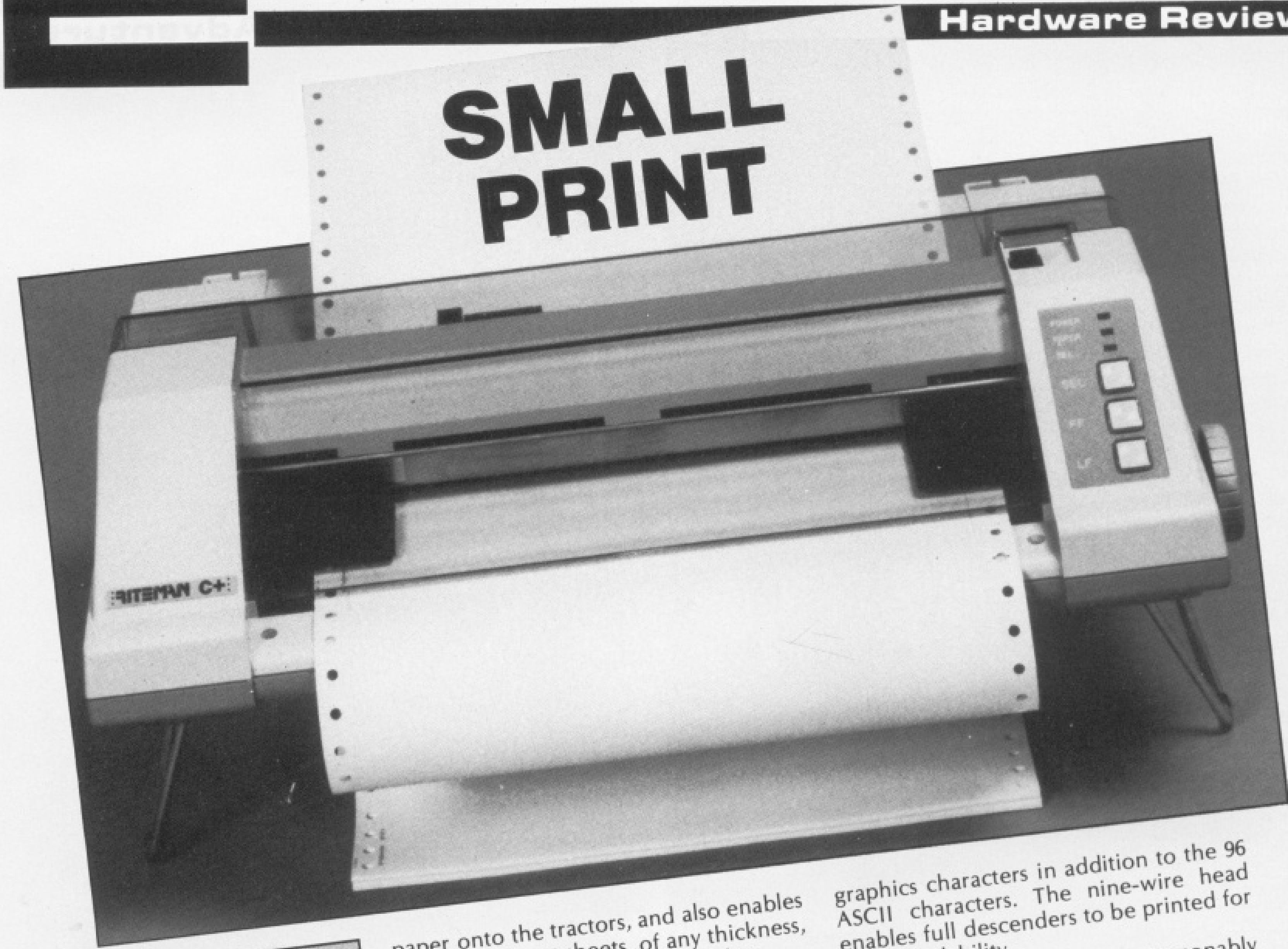
SAVE and LOAD allow you the option of 10 different game positions. These may be overwritten at any time you change your mind throughout the game. QUICKSAVE and QUICKLOAD are particularly nice features as they allow you to save your present position 'temporarily' at any time you think your next move may be your last!

HELP enlists the assistance of a wise old bird – the Condor – why a Condor I do not know but he can be darn useful! His help can only be given three times...so make use of his knowledge carefully. You can of course start from scratch and use the help gained in previous games to reach your last position, then get three new HELPs!

Although there are some devious parts to Mindshadow, it is not a very difficult game to play. The number of locations in each section is not large and although mapping should always be an adventurer's first line of attack it is not always necessary here. You must solve each section before you can proceed, so at least you know how you are doing! If you are unable to go anywhere else then you know you have missed something!

For all that, I still do not know who I am...there are no prizes to anyone making any unkind suggestions!

SMALL PRINT



Barry Miles has had a look at the C.Itoh Dot Matrix Printer — and gone overboard.

THIS PRINTER IS BEING SOLD AS A competitor to Commodore printers. It is fitted with Din sockets and an internal interface making it directly plug-compatible with Commodore machines.

In addition, it responds to Commodore's control codes and secondary addresses. It is as if the Commodore range of printers has been extended. You can also switch the machine into an Epson FX80-emulation mode. It is immediately apparent that serious thought must be given to this printer.

First impressions are very good. This is not a hernia-inducing machine. It weighs a mere 8.375 pounds and comes in a box with a nylon carrying handle, which emphasises its lightness. Inf act it could probably be carried in a normal briefcase.

Additionally, first examination of the machine shows that very careful thought has been given to its design. It is both a tractor and friction-feed printer, and has a carefully designed paper-transport system, which keeps the paper flat. This is a great advantage, since it permits extraordinarily easy front loading of paper onto the tractors, and also enables you to insert cut sheets, of any thickness, (even postcards), without difficulty.

Furthermore, the paper you are printing on does not get bent at all during the printing process.

Another good feature is apparent in the way in which the controls have been placed fully visible outside the case of the machine. You do not have to grope down into the bowels of the Riteman, looking for the lever to vary the force of impact. Instead the lever which does this is fully visible, and sensibly labelled.

Similarly, the paper guides are marked making it easy to set up normal listing paper or A4 accurately.

Switching between tractor and friction is a similar trivial task making the printer foolproof and a pleasure to use.

The simplest and cleverest innovation is the sturdy foldaway wire stand which lifts the printer just a few inches off the table top, thus making room for a decent-sized pile of paper.

You have only to trouble about the destination of the output. If your table extends backwards further than the back of the printer, you can allow the paper to build up there. Users who prefer to have the whole box of 1,000 or 2,000 sheets on the floor will find that when the stand is folded away, there is still clearance for the paper to pass beneath the printer. C.Itoh seem to have thought of everything!

The machine runs at 105 characters a second, and prints all 82 Commodore graphics characters in addition to the 96 ASCII characters. The nine-wire head enables full descenders to be printed for good readability.

In use the printer was reasonably quiet, and extremely easy to operate. The dip switches to make more fundamental adjustments are easily accessible once the single self-tapping screw holding the flap in place is removed.

The features which are likely to impress most of all are the Bit Image graphics at 480 or 960 dots per line, and the complete EPSON compatability. This means that all the fancy Control Codes can be invoked, giving double strike, emphasised, double emphasised, italic, compressed and expanded, subscript and superscript characters.

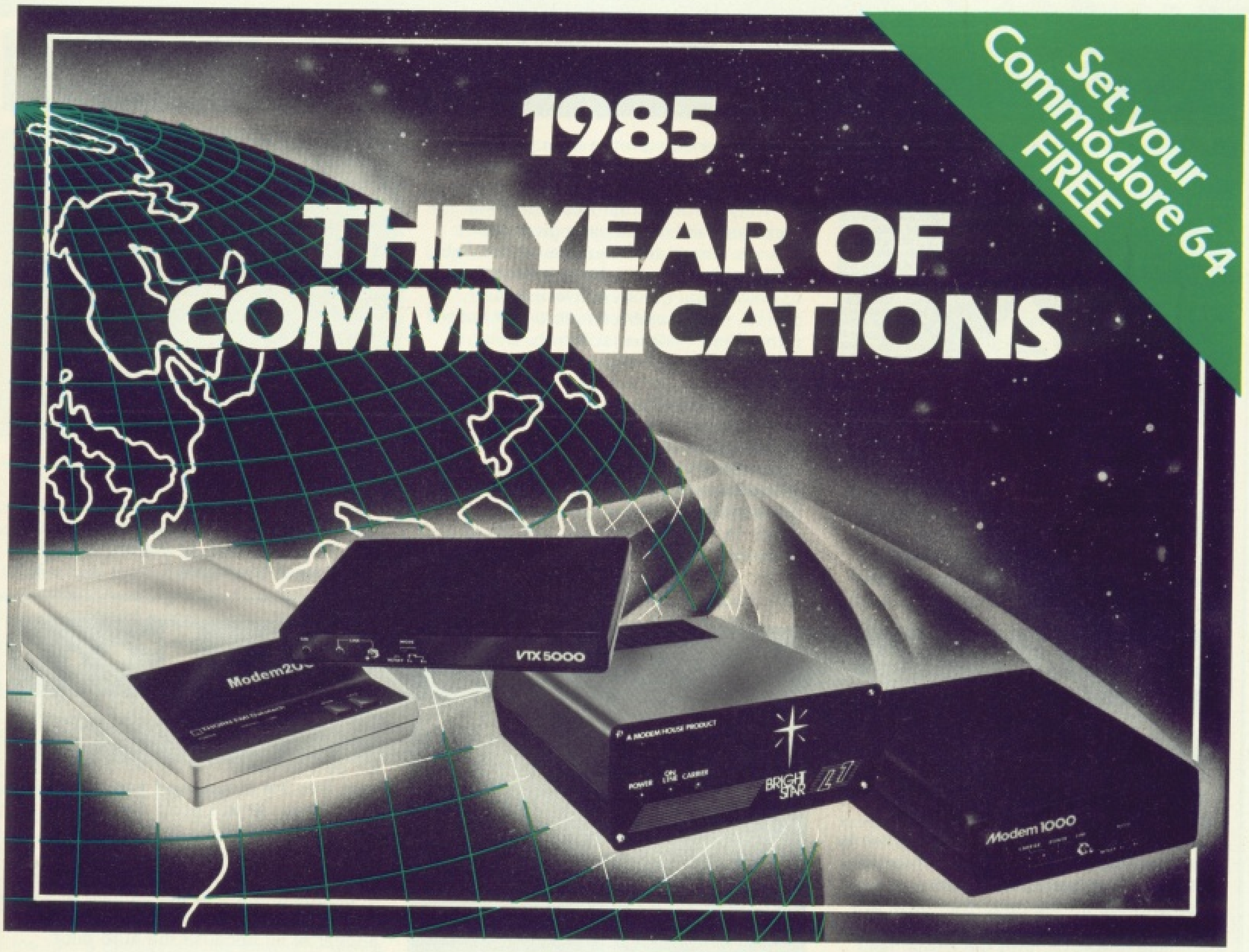
Near Letter Quality print, by means of multiple passes is not available, but as you will see from the sample printouts, all other likely requirements are catered for.

Indicator LED's show you when power is on, paper is out, and whether the printer is On-Line or not.

The Select button toggles the printer on and off-line, and the Form Feed and Linefeed buttons perform their obvious functions.

Ribbon changing is a doddle because the ribbon is in a small neat cartridge, and the printhead is changed, after 50 million characters, by simply clipping it on and off.

I found the printer to be the simplest I have come across, and free from vices. Highly recommended and very good value for money.



A whole new world awaits you and your 64. The world of Prestel, Micronet, Viewfax, Homelink, Citiservice, This is the world of product reviews, mainframe games, home banking, business information, commodity futures, armchair shopping, real time conversation, electronic notice boards, helpful tips, ICPUG, news, Telex-Link, Mailbox, Chatline, Gallery etc. This is the world of telesoftware, with programmes to download into your 64 and save to disk or tape. Many telesoftware programmes are provided free of charge or at nominal cost.

Dialog with over 600 massive databases on call and almost unlimited information on just about any subject is accessible from your 64, together with BT Gold, Easylink, One to One etc., with electronic mail, business services and telex facilities.

You can link to Bulletin Boards in your area for local news, club news, letters, jokes, poems, helpful tips, etc. It's a whole new experience.

Take the first step. Make a giant leap for your 64. Communicate with Modem House and you'll soon be talking to other '64 users, linking to other micros and swapping files, tips, programmes or just chatting.

Modem House are the largest suppliers of Commodore communication equipment. We also supply complete systems for the Pet 2000, 3000, 4000 and 8000 series.

MODEM HOUSE

Iolanthe Drive, Exeter, Devon EX4 9EA. Tel: (0392) 69295

The most popular 64 micropack in use today, will link you to Prestel, Micronet 800, Viewfax 258, Homelink, Farmlink, Citiservice, etc. From Modem House, the complete system is on offer at the incredible price of £99.95 inc. VAT, p & p.

Optional terminal software to link you to BT Gold, Easylink, One to One, Dialog and Bulletin Boards is available at £29.95 inc. VAT, p & p.

Complete Commodore 64 micropack

special price **£99.95** inc. VAT p & p
for a limited period only.

Special, special offer, order your 64 micropack now and get a **FREE** quarter's subscription to Micronet 800 and Viewfax 258 (Prestel Microcomputing).

Modem House, Iolanthe Drive, Exeter, Devon EX4 9EA. Tel: (0392) 69295

Please send me .... Commodore 64 Micropack(s) £99.95 each
.... Optional Terminal Software £29.95

I enclose £................../Bill my Access No. ....

Name ....

Address ....

....

....

Telephone No. ....

Please allow 28 day delivery

This offer applies only while stocks last

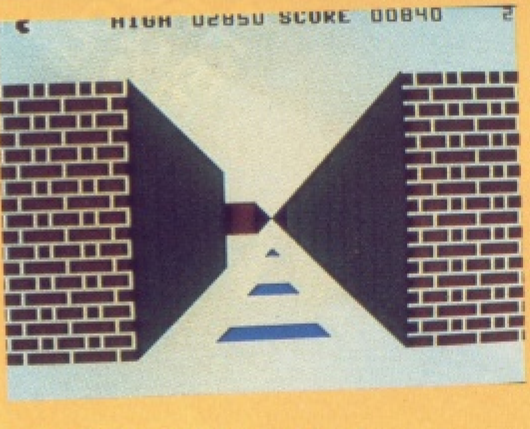
YC11/85

3-D GLOOPER

Supersoft £4.95

* * * * *

Originality 3 | Playability 6 | Graphics 7 | Value For Money 8



C16/Plus 4 Joystick optional

This game, originally on the 64 has now been converted to the 16. In effect, the game is 3D Pac-man and you are actually standing in the maze. The scenario is as before. You must collect all the blue dots on the floor of the maze whilst avoiding the attentions of the gloopers.

These brown creatures patrol the maze and will munch you given half a chance. There are a number of red dots which empower you, for a limited time, to munch the gloopers. During this period, the gloopers are coloured yellow rather than their usual brown. Clear all the dots and you move onto another screen.

The screen shows a perspective view of the maze walls and side passages. A combination of brick patterns and coloured shading helps the effect. The gloopers are quite simple in design but grow nicely in size as you approach them. In all a surprisingly neat bit of graphics.

You can examine your immediate surroundings on a small map which appears at the top of the screen. This shows the position of dots and gloopers. If a glooper starts to approach you, a warning sound is given. The keyboard or joystick is used to move about the maze. You can move at quite a speed but this isn't always advisable since you can easily turn a corner straight into the arms of a glooper.

In this world of more sophisticated software, it's nice to play a simple but addictive game. I found *Pac-man* rather tiresome since the whole action is visible. In this game, you have no idea where the gloopers are. This engenders a surprising amount of excitement. I found this game great fun to play and quite addictive. The graphics are simple but most effective. At the price, quite good value for money.

M.W.

SKYFOX

Ariolasoft £9.95

* * * * *

Originality 8 | Playability 9 | Graphics 9 | Value For Money 9

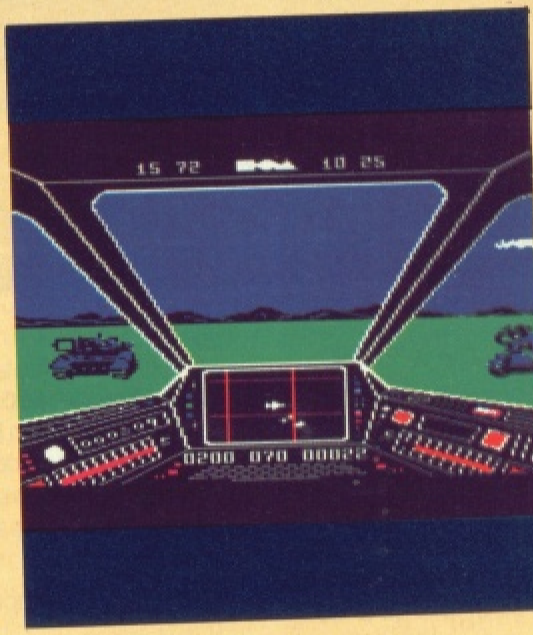
C64

As its title Sky Fox might suggest it is about a jet fighter but it's not one of the flight simulators that have become popular lately, more of an air combat simulator. Thus the pilot doesn't have to concern himself with all the intricacies of take off and landing but can concentrate on the important business of zapping baddies.

This brings us onto the game itself. The story goes that the baddies have invaded, either from another country or another planet, and are about the usual invader – like things such as killing and destroying. Your job is, of course, to repel the invaders and save the world/country. In order to achieve this you are equipped with the Sky Fox fighter plane, a weapon capable of standing still in the air or zapping along at speeds of 3000mph! The Sky Fox boasts a full complement of baddie blasters ranging from general purpose laser guns to the air to air heat-seeking mega missiles.

The screen displays a cockpit view that has three large windows and a radar display which shows either an all round proximity scan or a sort of weaponry screen complete with gun sights. Also available is a direct link to the main battle computer at your base. When activated this swings out on an expanding window to cover the whole display. This computer link supplies information about the progress of the war with the status of various installations being displayed so that you can judge where you are most urgently required.

Also in the data banks is a battle plan displaying the position and number of your foes as well as the locations of your base and the other establishments. This plan may be used to navigate but



using the auto pilot is much more fun as it will either take you to a specific location or it will find the nearest zappable baddie and zoom there at 3000mph.

Whilst being great fun and very convenient, this auto pilot knows nothing of air combat and thus usually engages you with the enemy from the worst possible position. Combat takes two forms. First is the ground force consisting of groups of five tanks. These appear on the display as tiny little blobs on the horizon. At this stage the blobs may be disposed of with the lasers very easily. This would be a shame however because the blobs grow into very detailed battle tanks that shoot at you as you approach and to miss this would spoil the fun. So my advice on the tanks would be destroy all but one, fly over that admiring the excellent graphics and then turn round and blow it to hell!

The other form of combat is in the air, as you climb from ground level you pass into the cloud layer and the screen goes blank. At this point the disk is being accessed but I think the cloud idea is a nice cover for this. Once through you are presented with a view of clouds and are in the domain of the enemy jet fighters. These usually don't appear at all because the auto pilot lands you in the fight with the nasties behind you. If you do manage to get one in front of you don't forget to be amazed by the smooth animation of such a large and detailed shape before you target the heat seeker and despatch it to the same fate that the tanks suffered. This aside, I should state that the air combat is not easy at all and took me many goes to get the hang of it.

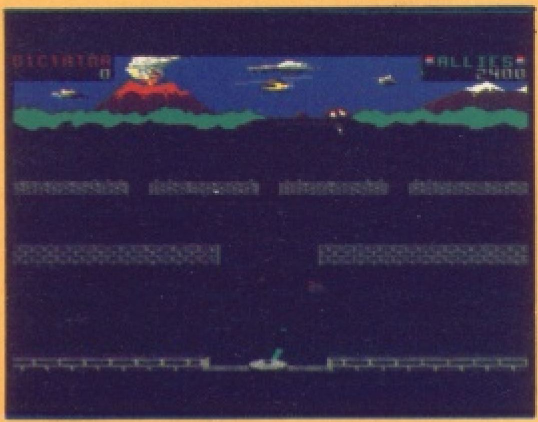
The biggest and baddest of all are the mother ships. These are like large floating cities in the sky. The mother ships send out all the other baddies so a good plan is to take on the mother ships first to prevent them sending out replacements for the tanks and jets that you destroy. As big as they are they can be taken out by laser work but a lot of careful flying is required to get past the internal defences and they need to be hit a number of times due to their armour. Once you have exhausted your supplies of missiles and fuel you can land at your base to stock up. That is if it still exists!

The game has a large range of options such as difficulty levels from Novice to Ace of the Base and practice levels involving only air or only ground combat. Also selectable is the scale of the baddies' attack and the form of your task. These have titles such as cornered, The Alamo and, my personal favourite, Massive Onslaught.

BEACH-HEAD II
US Gold £9.95

★ ★ ★ ★ ★

Originality 8 Playability 6 Graphics 6 Value For Money 9



C64 — 1 or 2 joysticks
This sequel to the successful Beach-Head is not just one game, but four games in one, for one or two players.

The evil dictator known as the Dragon has captured some of your men and is holding them hostage in his heavily fortified island home. The task of the allied forces is to storm the island, rescue the prisoners and escape with them by helicopter. You then do battle in single combat with the Dragon himself.

These four screens may be played in any order, with three skill levels for each. You may choose to play the allied troops or the dictator's forces, so many permutations are available.

The attack phase involves landing your troops by parachute from a helicopter, then advancing them individually past two barriers under withering machine gun fire. You will lose many of your men, and will need to sacrifice some for the safety of others. Bonus points are scored for successful grenade attacks on the machine gun position.

To rescue the hostages you use a captured machine gun against a bewildering array of forces, all intent on killing them. This is not easy, as the tanks and armoured trucks need to be hit in exactly the right spot to knock them out.

In the escape stage you fly helicopters, loaded with escapees, past a series of obstacles while under constant fire. You can strike back, and you score points for targets destroyed as well as soldiers rescued. The Dragon sets the difficulty level for each of the three sorties, and the highest level is really hard!

Finally, having tracked the Dragon to his lair, you engage him in battle with wooden spears called Poontas, from opposite banks of an underground river. You move, duck and jump, and can control the spears in flight, making them swerve. Four hits win a round, nine of which must be played in all.

A marvellous game - challenging throughout, with excellent graphics and the best software speech I've heard. It deserves to be a great success!

P.R.B.

PAINTBOX
Audiogenic

★ ★ ★ ★ ★

Originality 2 Playability 9 Graphics - Value For Money 9

C-16 - joystick optional
For those who like graphics without mathematics, *Paintbox* is a must for Commodore users; I say users as one side of the tape is for the C64 and the other for the C-16.

Loading is fast and efficient and the menu display scrolls smoothly along the bottom of the screen. At any point the menu is instantly accessible by pressing the space-bar. The menu controls a variety of options including saving to tape or disk and a fast erase.

Now what does the program offer? DRAWING is, of course, a standard feature using joystick or keyboard. To this is added:

Line - Allows a line to be drawn at any angle.
Lines - draws a series of lines starting from where the last one finished.
Pays - gives a series of lines starting from the same point.
Framing - provides a rectangle of any given size at any position on the screen while BOX gives a solid rectangle of any chosen colour. CIRCLE at any place, any size and use DISC to give a colour filled circle.

This sounds pretty much the same as usual but, believe me, the use of 'rubber-banding' technique gives you tremendous scope and versatility.

When we come to PAINTING the full colour range is available for filling enclosed areas while, in DRAWING mode eight options are available from very fine line to very thick pen which are really excellent. Drawing speed may also be controlled.

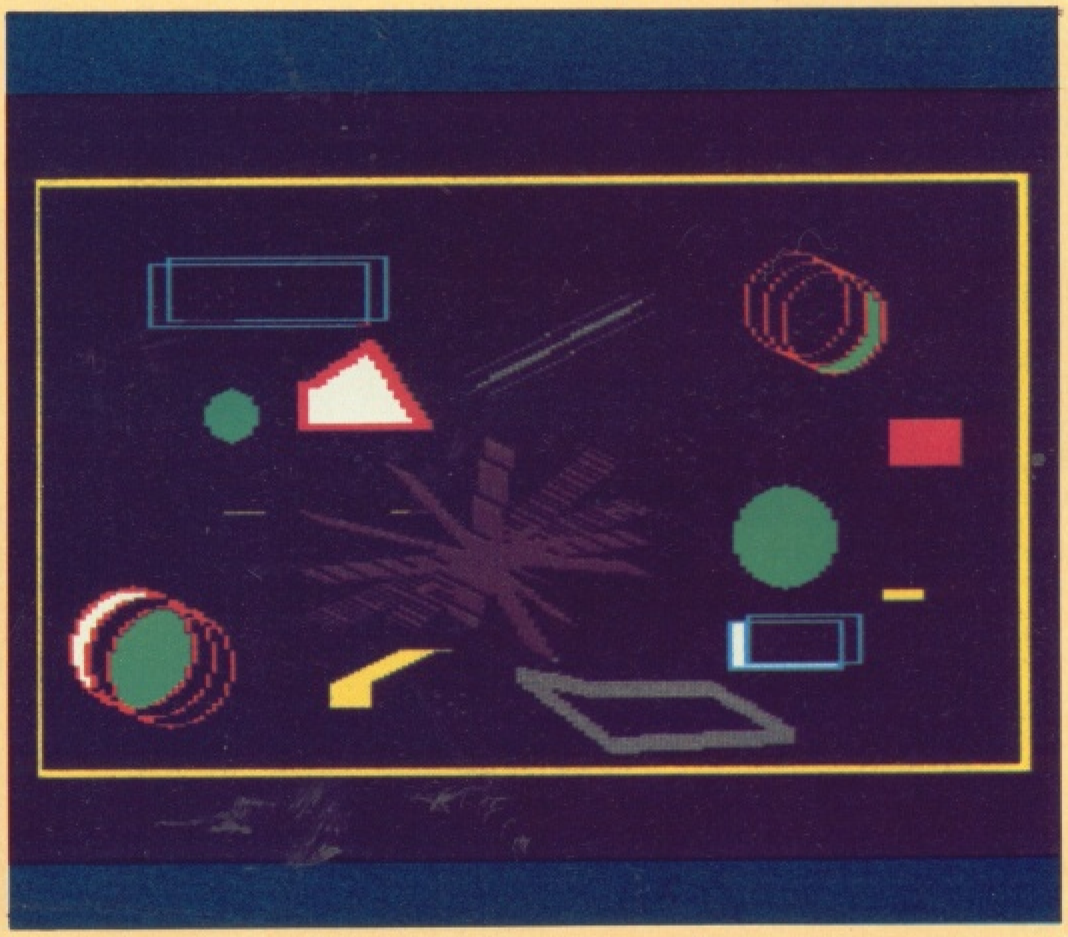
For the C64 alone there are extra facilities allowing you to copy, SWAP and MOVE two pictures around building one upon the other.

Audiogenic has produced here a very good, fast acting, easily handled tool which can be used to create very intricate drawings, picture drawings, poster drawings etc.

Paintbox compares most favourably with other graphics utilities and, indeed, would be the one of my choice by virtue of its speed, scope, ease of handling and well written manual; screen colour control is likewise excellent.

A joystick is preferable but keyboard control is fully functional and adequate.

Go to it beginners and advanced artists. Well recommended. **E.M.**



SPEED KING

Digital Integration £9.95

★ ★ ★ ★

Originality 3 · Playability 7 · Graphics 7 · Value For Money 5



C64 — joystick optional

If you have quick wits, cool hands and an iron nerve, then this is a game you must buy! You can experience the thrill of high-performance motor-cycle racing at any of the world's top 10 circuits, from Brands Hatch to Daytona, at speeds of up to 250 miles per hour. And all this from the comfort of your own armchair!

Each of the circuits is faithfully reproduced, with good graphical representation of the scenery, the track and, of course, the bikes. You race against 19 other riders, starting at the back of the grid. If you collide with another bike you crash, but can set off again. Wandering off the roadway slows you down too, and eventually makes you lose control.

Your machine is equipped with six gears, which you can change up and down, and the sound effects accompanying gear-changes are most realistic. Holding down the fire button (or space bar) causes constant acceleration. Releasing it slows you down, but for heavy braking you will need to change down.

Various options are open to you. In addition to the choice of circuit, you may select novice, pro or champion skill levels, and you are given a preview of the chosen track. You may also do as many practice laps as you wish, before setting off under race conditions. The length of the race may be two, four or six laps. The computer records the total time for the race, and your fastest lap time.

I was impressed with everything about this game. It is very well programmed, fast and exciting. There is plenty of scope for improving your skill by practising, and plenty of variety. It sets a standard for motor-cycling games which will be difficult to beat. So put on your crash-helmet and get started! **P.R.B.**

RACING DESTRUCTION SET

Ariolasoft £14.95

★ ★ ★ ★

Originality 9 · Playability 8 · Graphics 7 · Value For Money 9

C64

Have you ever had the urge to get in a powerful car and race round a track as quickly as possible? Destruction Set will allow you to indulge all your fantasies and more!

The game consists of two race cars on the same track. Each car's progress is displayed on a separate window on the screen so if one driver gets ahead of his opponent he doesn't disappear off the screen. The cars are rather like slot cars as they will turn corners on their own and the only control required is to change slots and accelerate.

So what is the Construction in the title for? Well this is because RDS allows the user to either choose one of about 20 pre-formed race tracks or even to make one up himself. This allows all sorts of differing terrains and thus the game can change from a Formula One at Silverstone to a dirt track rally on the moon!

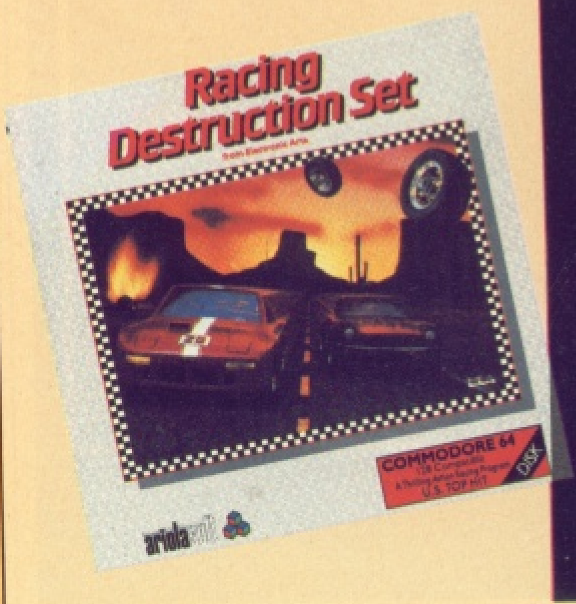
To enable the cars to race on the different surfaces the user can also control the design and specification of his car. This allows such things as a Corvet Stingray with an 8.2 litre engine right down to a 250cc Bike.

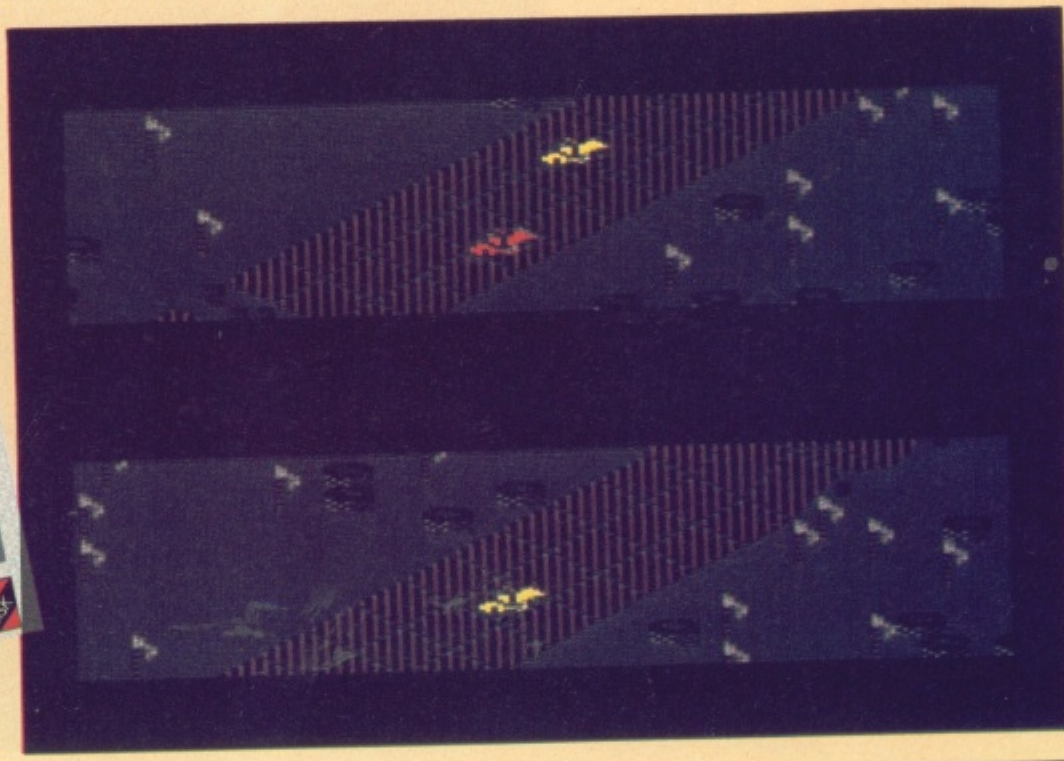
Care has to be taken when choosing a vehicle as it is very easy to construct a car that is too heavy to get up the steep hills or not powerful enough to get off the ice. It is great fun however to pit two entirely different cars against each other on a variable track and see one catch up on the straights due to its power but then fall behind on the hills due to its weight.

What about the destruction bit, I hear you ask? Well this is great fun as well because the game offers options to carry oil for throwing in the path of your opponent and land mines to blow him up! The problem with both the oil and the mines is that they stay where they are and stay nasty as well so if you drop a land mine in a place where your opponent can't miss it he will blow up, but on the next lap if you can't get past it, so will you!

Although the game does a lot of very slow disk accessing, I think that it's excellent and I recommend it.

J.G.D.



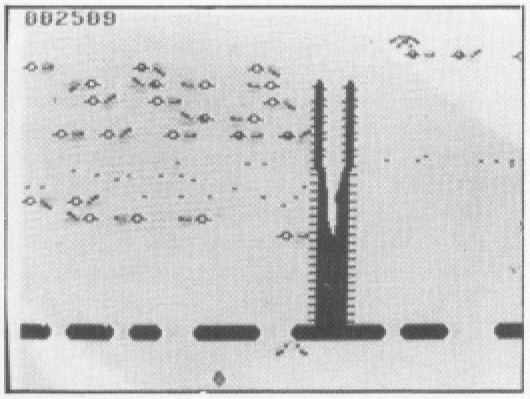


KAKTUS

Supersoft £4.95

★ ★ ★ ★

Originality 2 Playability 7 Graphics 5 Value For Money 7



C16/Plus 4 — joystick optional
This is another C64 conversion from Supersoft. From memory, I don't think have been many alterations from the original.

The plot is somewhat unusual in that you must protect a cactus from the unfriendly attentions of a swarm of wasps and hornets. Your efforts are hindered by interfering moles and vultures. In effect this is a shoot-em-up game similar in flavour to centipede but with its own attraction.

On screen you get a side view of the cactus and the ground level. You may move above or below the ground via holes in the ground. The swarm of wasps and hornets approach from above and fly from side to side slowly moving downwards. You must prevent them from reaching the base of the cactus where they can do the most damage.

Unfortunately, it isn't a simple matter of shooting the wasps. Moles move from side to side trying to fill in the holes in the ground and cut off your means of access. You, of course, can shoot them. After screen one, vultures fly across dropping eggs. These bounce about and will destroy you on contact.

Graphically, the game is simple with limited use of the C16's colour abilities. The animation of the wasps is quite good but overall the nasties don't really look like the creatures they are supposed to be. Sound is limited to zapping noises although a nice version of Lincolnshire Poacher starts the game.

Whilst this game doesn't break new ground in sophistication, it presents a real challenge. The first screen is deceptively easy, but our friends the vultures make life pretty tricky later. They seem to aim their eggs accurately and the bouncing bomb behaviour of the eggs is tricky to judge.

Overall, not a bad game especially bearing in mind the lack of material for the C16. The rather primitive graphics do the game no favours but it does offer a good old fashioned zapping session.

M.W.C.

WORD PERFECT

Supersoft £19.95 disk

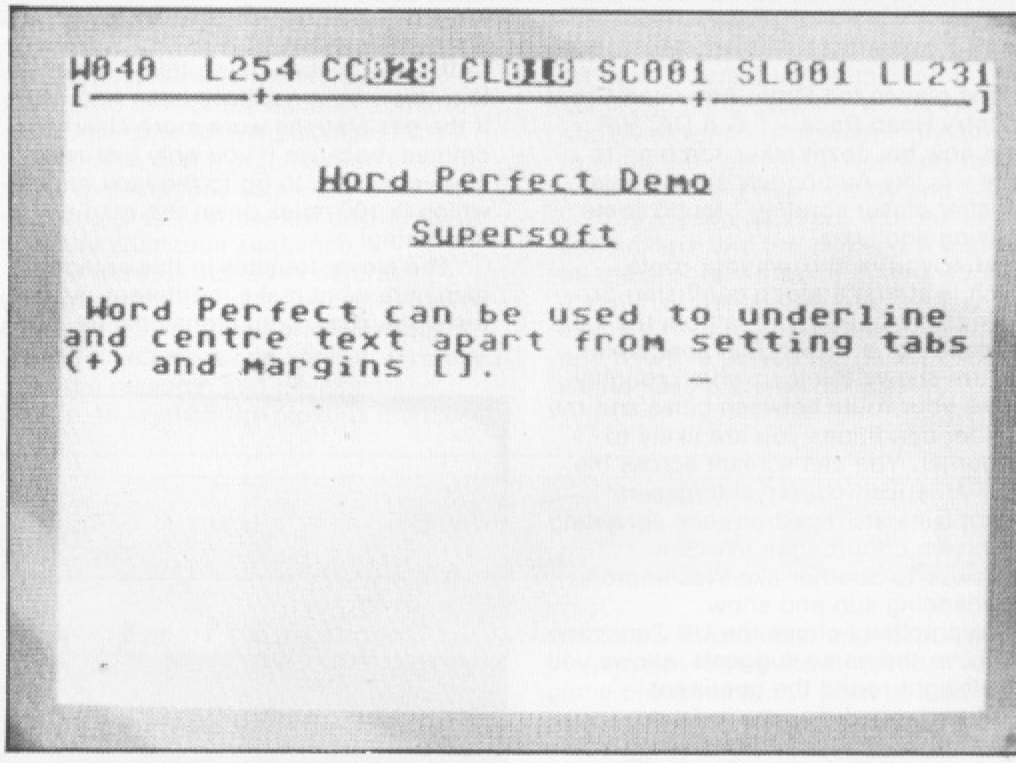
★ ★ ★ ★

Originality 1 Playability - Graphics - Value For Money 9

C64
I dare say that a good many owners of the 64 have considered the benefits to be derived from using a word processor but have been deterred by the high price of most of the programs currently available.

Even if you don't own a printer, a word processor can still be extremely useful. How many of you, for instance, can sit down to write a letter and get it right first time without having to rearrange paragraphs, correct spelling or syntax errors, or even re-write the entire thing, perhaps several times? With the aid of a word processor and a little typing practice the whole task becomes very much easier. Using the screen as a sort of electronic note pad, you can organise your thoughts and correct any errors, all without committing anything to paper.

However, whilst appreciating the above, many people will be unable to justify the expense of a program which although very useful is somewhat limited in its application in the home. *Word Perfect* from Supersoft has recently been released and seems to be aimed at just such a market. The program is available in both disk and tape format, the disk version being the one reviewed here.



Word Perfect is a scaled down word processor and as such it is suited to any of the uses to which you would put a typewriter, but with most of the advantages of word processing. The program loaded quickly and reliably and resides in the protected RAM above Basic starting at 49152. Most of the important features of a word processor are there, although some of them such as block move and copy are implemented in a somewhat crude fashion. You may search for a specific word or phrase, centre, underline, set tabs and merge previously saved text into the current work. You have the option of saving your finished document to tape of disk.

Although some of the more usual commands are missing, right justify and search with replace for instance, I don't feel that their absence will cause too much inconvenience.

The program allows the use of a wide range of printers although, as with all software tht uses a printer, it would be prudent to check that your particular set up works before buying.

Word Perfect comes complete with a well written 16 page booklet that should enable even the novice to quickly get to grips with the program. While perhaps not suited to serious office work, I feel it has a lot to offer the home user. This review was written with the aid of *Word Perfect* and I shall continue to use it in the future.

My copy of the manual has an error on page eight. To reformat a paragraph you should press DELETE in CONTROL Mode, not R as stated.

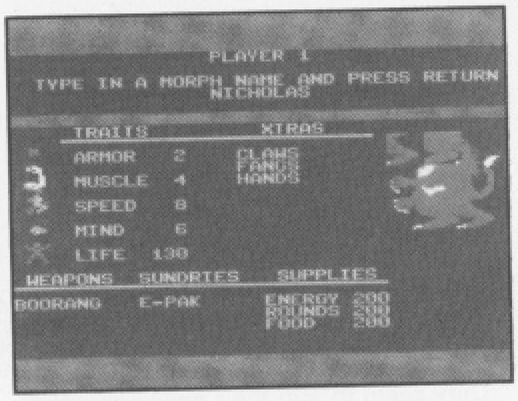
D.J.T.

MAIL ORDER MONSTERS

★ ★ ★ ★

Originality 10 Playability 9 Graphics 6 Value For Money 8

Ariolasoft £14.95



C64 + joystick

Although the title is different, this is one of the excellent Electronic Arts 'Construction' series.

The game starts by telling you to select a Morph i.e. a basic body type. This varies from the Human Homin to wasps and strange lion like creatures.

Next, weapons must be selected to arm your particular beast. These range from laser rifles and grenades for the humans to noxious spit from the worms and of course deadly claws from the lions.

Then the creatures travel to the teleport chamber where they are left whilst the controllers decide what type of combat is to be engaged. One owner will decide the terrain, and this should be chosen to give your creature the maximum advantage and your opponents the least. So if an owner has an amphibious creature he could choose a terrain with lots of water thus confining his opponent to the land.

The other owner must try to nullify the disadvantages of the terrain by selecting the type of combat. The first type is the simple destruction combat where each player will fight to the death with the creatures found in the arena and the opponent.

The second requires flags to be collected in order. Each flag is defended by an arena creature.

The third type is called the horde. This is when both creatures must work together to defeat an invading horde that comes from the top of the screen and is attempting to reach the bottom. The one who kills most hordlings wins.

As I mentioned, the arena contains 'Urban Defenders'. In a one player game all these creatures are played by the computer but in a two player game the creature is played by your opponent. So, if a creature runs into an urban defender, the screen clears to a combat screen and the two players battle to the death. This is done by selecting weapons and blasting the enemy. This requires a great deal of dexterity and practice is recommended. If the player is victorious in his attack on the urban defender then the screen changes back to the large map and the game goes on.

Graphics on this game aren't stunning but none of the construction series are. The game is a different matter as it is excellent and I found it extremely addictive. **D.G.D.**

THE GREAT AMERICAN CROSS COUNTRY ROAD RACE

Activision £9.99

★ ★ ★

Originality 4 Playability 7 Graphics 7 Value For Money 7

C64

At first glance the Great American Cross Country Road Race – T.G.A.C.C.R.R. from now on, coz it takes too long to write – is like Audiogenic's Tallegeda, but after closer scrutiny I found some pleasing additions.

After you've chosen your route, which is at first a touch confusing (it reminds me of trying to fathom the intricacies of a British Rail time table) – you are shown the map which roughly shows your route between cities and the weather conditions you are likely to encounter. You can scream across the great American countryside/desert/snow plains/etc. etc. from one sprawling downtown conurbation like San Francisco to another like Washington, experiencing sun and snow.

To practice I chose the US Tour race which, as the name suggests, allows you to belt right round the continent.

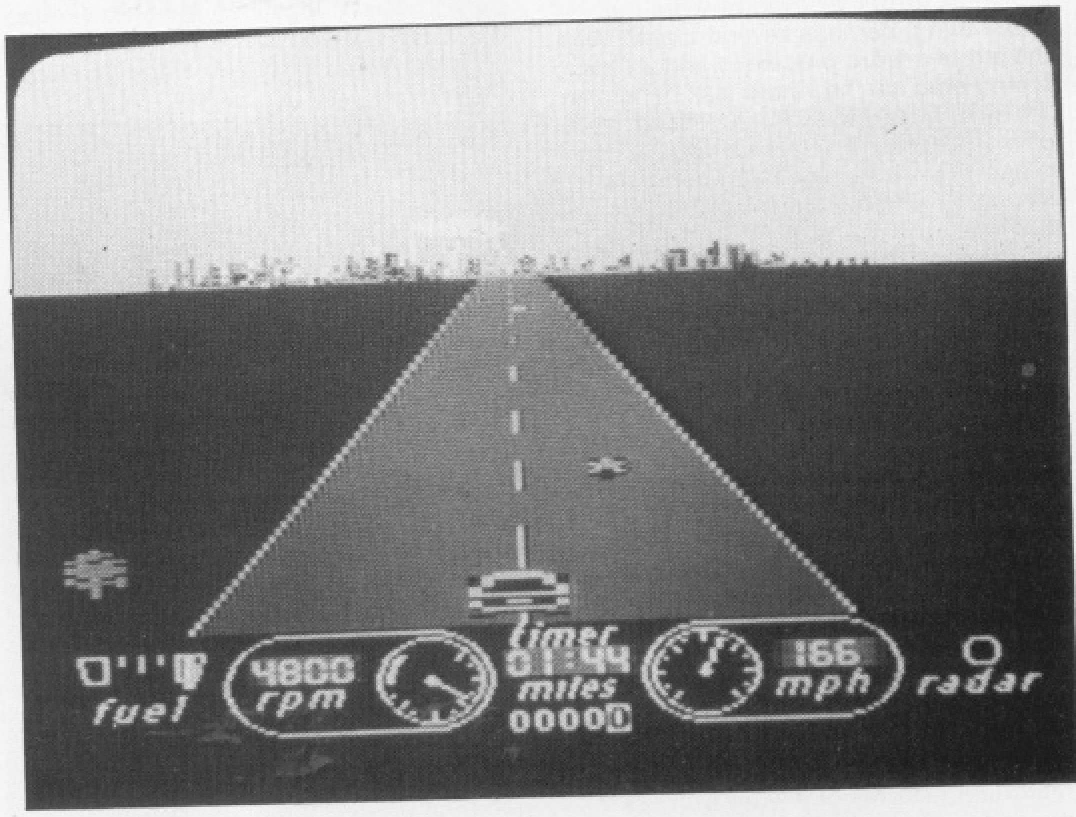
After the prelims, you're faced with the old hat track, down which you must guide you low strung speed machine.

You have a certain amount of time to get from one check point to the next and you're expected to change gear using the joystick, while keeping an eye on your fuel. If you allow your RPM to stay in the 'red-line' too long your engine blows up but it doesn't take too much skill to avoid this catastrophe.

Fuel consumption is, I found, another story. This particular hazard really wound me up. When you run out of gas you have to "push" your car to the next station. You do this by continually pressing the fire button – I actually broke my joystick doing this. I believe that this would not have been necessary if the gas stations were more clearly defined, because if you only just miss one you've got to go to the next one which is 100 miles down the road – infuriating!

The clever touches in this arcade game are what make it different. As you travel the road, conditions change along with the scenery and it gets dark – I liked that. If you go too fast you attract the attention of the fuzz and they pull you over.

Altogether this is just another road-race game and, although clever in places with good scrolling graphics, I feel that once you've seen one there isn't a lot of difference in the rest – but this is certainly one of the best. **D.F.**

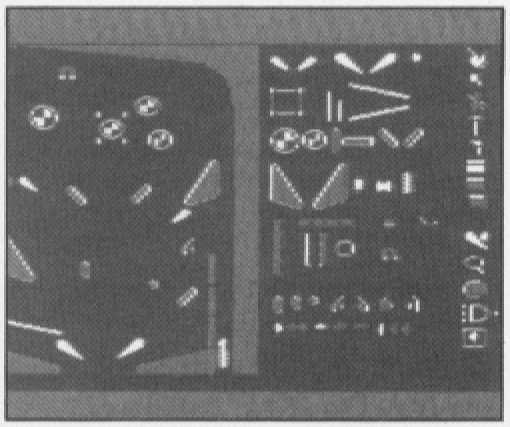


PINBALL CONSTRUCTION SET

Ariolasoft £14.95

★ ★ ★

Originality 7 | Playability 7 | Graphics 8 | Value For Money 6



C64 + Joystick

The Idea behind the construction series of games is that people could adapt a game which they liked to take on many different forms and levels of difficulty so as to improve their enjoyment. This isn't the case with Pinball C.S.

The game is a representation of a pinball board which can be set up to the user's requirements. All manner of bouncers, flippers and fancy bits may be put together on the board in an attempt to find an interesting game. This is where the construction bit falls down as most fun is to be had in the construction of weird and wonderful boards. Playing these boards soon becomes very boring indeed! As all the fun in real pinball is in the flashing lights and ping, ping, ping noise.

Anyway, back to the game. As I mentioned, the game allows use of bumpers and flippers. It also has many advanced features such as a ball hopper which collects balls, until it's full at which point it releases them all putting four balls in play at once. Another interesting feature is the AND gate that allows special scores to be awarded for good play.

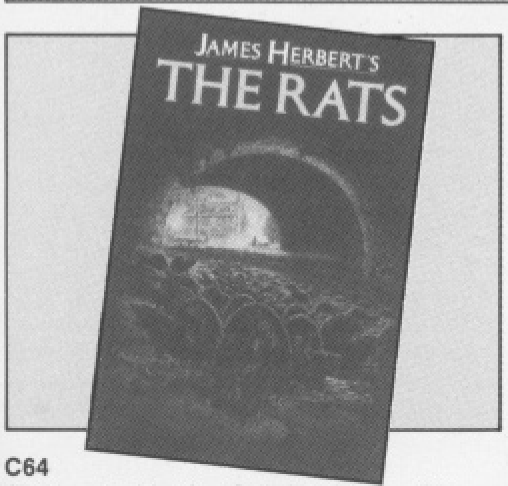
All in all I was disappointed by this. Nice idea but a boring game. **J.G.D.**

THE RATS

Hodder and Stoughton £7.95

★ ★

Originality 9 | Playability 3 | Graphics 5 | Value For Money 4



C64

As reported in the October issue of Your Commodore, Hodder and Stoughton Software has released the computer version of that now age old horror story The Rats by James Herbert.

When I got my grubby little paws on the game I could hardly wait to get it in the C2N and have a look - you see, dear reader, I'd read the book and seen the film already and thought they were both terrific - so maybe I expected too much.

In the game you take the roles of various characters in the original story - except the rats, of course. You can, for instance, be anyone from the controller of defence to a cobbler which, altogether, adds to the general confusion, and this, in my opinion, is the only atmosphere generated by this attempt to mix all the aspects of micro-gamesmanship in one package. I think it's a shame that the game didn't live up to my expectations especially after the stunning graphics that are displayed after the animation sequence.

The game is staged in central London where the rats are swiftly taking over and it's your task, with the help of the defence forces and various wizardry from the research and development people, to contain the hideous menace within London, whilst keeping the three main characters alive and not at the expense of every last man jack in your defence forces. To fail in any part results in defeat and some supposedly horrifying graphics.

To effect the destruction of the rats you are given various tools and professional forces to help you and you must put into action Foskins (one of your main characters) strategic battle to do this. This is done by moving a cursor again to pick the words (your choice of which is maddeningly limited) to build the command. Needless to say, I was shredded very quickly and often.

It is a good idea, but with the lack of atmosphere and the limitations imposed by the program I found it hard work and it lost my interest very quickly. Like I said, I'd read the book, seen the film, and now I've played the game. I think I would have prefered the pie!—

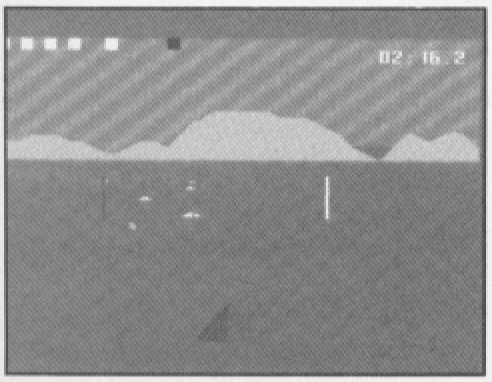
D.F.

KAYAK

Creative Sparks — Sparklers £2.50

★

Originality 2 | Playability 5 | Graphics 3 | Value For Money 3



C64 — joystick

If you find it exciting to have a cold bath on a foggy Sunday afternoon in November, while Chopin's funeral march plays in the background, then this is the game for you! According to the rather ungrammatical cassette insert, your mouth will be dry and your arms shaking with excitement — lies, all of it lies!

The idea of the game is that you paddle a canoe in the World White Water Slalom Championship. This involves two forays along a stretch of winding river, with 25 gates to negotiate, remembering to keep the white pole to your left at each. The control you have over your craft is minimal. Attempting to turn usually ends with you ramming the bank, at which point the program is as likely as not to seize up altogether.

The "superb scrolling 3D" graphics, mentioned on the packaging, are in fact crude and jerky, with the prow of your boat bobbing up and down in the foreground and slalom gates appearing, as if by magic, just in front of you. Making headway against the current is practically impossible, and response to the joystick is very slow indeed. The game proceeds at the pace of a snail with corns!

The only sound-effect, apart from a discordant fanfare at the start, is like nothing so much as brushing a pair of shoes, or perhaps a veteran washing-machine. There is a high-score table, of a sort, but even that is mud-coloured. The inlay asks "Will you have the stamina and strength to last the whole course?" - I very much doubt it!

If I seem rather unkind, it is well deserved. This is the poorest game I have seen for a long time, and the best thing to do with it is cover the tabs on the cassette and record something else over the top. Unless, perhaps, you suffer from insomnia?

P.R.B.

Pet

This month Margaret Webb explains how adventure games can be used as a valuable educational aid.

THIS MONTH I INTEND TO DISCUSS THE value of adventures as educational aids. In contrast to the monthly adventure columnist, with whom I don't want to compete, I will be advising the uninitiated, not preaching to the converted.

Many parents feel that programs other than educational software cannot be of any benefit to their children. This is simply untrue since we learn something no matter what task we're performing. The noisiest shoot the alien game may seem pointless but it does teach eye-hand co-ordination and helps improve reaction times. Adventures are a more subtle means of combining enjoyment with learning.

Before diving into the material, it would be best to try to describe what is meant by an adventure. Initially, adventures were simply word-games programmed on main frame computers. With the development of home micros, the games have changed form. First the text acquired graphics, then animation and interaction was developed and finally arcade/real time features appeared. All offer similar benefits albeit in differing proportions.

Most adventures involve placing the player, or his alter-ego, in an artificial scenario. By use of various forms of instruction, the player endeavours to solve certain problems or achieve certain goals. The scenario can basically be anywhere and can be mapped on a piece of paper. A building, for example, can be drawn in terms of rooms and passages.

The first task usually attempted in an adventure is to map, by exploration, the area where you find yourself – just as you would in real life. This means that you may need to know such concepts as the points of the compass and left, right etc. The scenario is normally too complex to memorise, forcing you to draw some form of map. Again this is a tricky task involving the manipulation of spatial concepts.

The mapping operations and the actual solution of the game need a rather special form of skill – the ability to think logically and apply lateral reasoning. A simple example is how to get past a fire breathing serpent which blocks your way when you're carrying a sword, a bucket of water and wearing running shoes. You could: Kill the dragon; put his fire out with the water; out run him; or use your map and find another route.

The problem is to find the correct solution and the means to achieve it using thought, trial and error and patience. All useful skills for real life.

Most of the best adventures use text to convey information and accept instructions. These processes will help improve reading, increase vocabulary, spelling, comprehension and creative writing but not to an extreme degree.

That's the boring bit out of the way, let us look at what's available. It might be easiest to deal with them in rough age groups (this list is not exhaustive, simply a guide).

Under-tens

The Magic Sword (Database Publications) is a fairy tale book plus a simple adventure. The adventure uses simple text with graphics and the instructions are mostly key commands. You must find the prince (who has been turned into a frog) so that he can save Princess Poppy from the wicked witch.

Danger Mouse in the Black Forest Chateau (Creative Sparks) is a graphics/ text adventure with menu driven command structure. There are lots of silly puns and great fun, pretty too. Help Danger Mouse save the world from the Pi-beam.

Early Secondary (and Smart Little-Uns)

This age group seems to prefer arcade type action and there are a range of tough graphics-only games to choose from. All of these require problem solving and lateral thought. Some worthy titles include: *Impossible Mission* (Epyx), an arcade style game requiring both physical and mental agility; *Staff of Karnath* (Ultimate), a magic type game involving exploration of a tomb and the fighting of foes with magic; *Elidon* (Orpheus) in which you guide the fairy on her search for magic potions and plants; *Dungeons of Ba* (Quicksilva), search the dungeons, find the treasure kill the nasties; *Shadowfire* (Beyond) uses advanced graphics techniques. You control an inter-galactic 'A' team trying to rescue an ambassador; *Tir Na Nog* (Gargoyle), a Celtic legend in which you help the hero find the Seal of Calum; *Exodus: Ultima III* (US Gold), Dragons and Dungeons style, you lead your band of heroes on a quest.

Teens to Old-Uns

The most challenging games tend to be text or text/graphics adventures. Whilst graphics help brighten up a game, you should be aware that the pictures only rarely give clues to the solution of the game. The list of such games is endless. There are some excellent software houses who are almost guaranteed to produce excellent games.

Those of note are:

Infocom – disk only, heavy use of detailed and absorbing text – a sense of humour vital. Those worth looking at are the *Zork Trilogy* (wizards and dungeons), *Starcross* and *Suspended* (science fiction) and *Hitch Hiker's Guide to the Galaxy* (humorous).

Level Nine – early games text only, later games use graphics. Complex games with many locations – text compression ensures detailed descriptions.

Adventure International – not the most complex games but have a very high content of puzzles. Great variety covering many types of scenario. Later games include *Gremlins* and *Incredible Hulk*.

Individual games of note are:

The Hobbit (Melbourne House) – on the book, quite tough.

Twin Kingdom Valley (Bug Byte) – superb graphics, complex and quite tough.

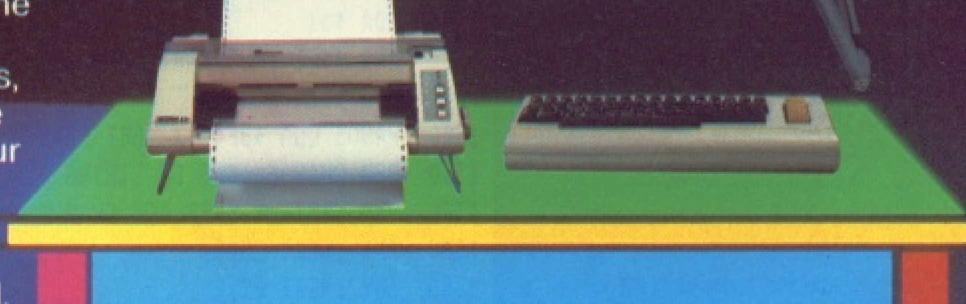
Fourth Protocol (Hutchinson) – amazing graphics, needs deep thought and a devious mind.

RITEMAN C+

THE ULTIMATE ENHANCEMENT TO YOUR COMMODORE®

- 105 cps uni/bidirectional logic seeking
- Supports Commodore 82-character graphics set, plus 96-character ASCII with true descenders
- Ten print modes, including standard, bold, expanded, condensed, emphasised, italics and super/subscript
- Commodore 64 and VIC serial interfaces
- Four international character sets – USA, UK, France and Germany plus Spanish and Scandinavian options
- 4″ to 10″ wide paper, traction or friction drive
- Super quiet
- One million character ribbon life
- Plus Epson® code compatibility





Advanced design features unique to the Riteman C+ give you big, fast, printer performance with Commodore graphics, in a super-compact, ready to use, portable desktop unit – making it first choice for your Commodore system.

A revolutionary front-loading design eliminates paper feed problems, and any weight of paper – even card – can be used.

The Riteman C+ also gives you a super-compact printer footprint. A built-in collapsible stand enables all paper to be stored under the printer, giving you 50% more desk space.



C. Itoh Electronics Co. Ltd.,
Beacon House, 26/28 Worple Road,
London SW19 4EE.
Tel: 01-946 4960. Telex: 8955616.

Commodore is a registered trademark of Commodore Business Machines

Epson is a registered trademark of Epson Corp

I want to know more about the Riteman C+, please rush me a datapack.
(Please indicate: End User☐ Dealer☐ OEM☐)

Name: ____________________

Company: ____________________

Address: ____________________

____

____

Build a better BASIC

Nick Hampshire adds more commands to your ever-increasing Basic.

IN LAST MONTH'S ISSUE I gave all the initialisation and wedge routines needed to add extra commands to the Basic of a C64 computer. Also included in that article was a single command CTL. In this second article I am giving four new commands, APPEND, CHANGE, DUMP and FIND. These are very useful 'toolkit' type commands for editing a program and are consequently all used in direct mode.

These four new commands all require the wedge and initialisation code – given last month – to be present in memory at the correct locations and that their command names and entry points are stored in the correct tables. It should also be noted that all these commands use common routines within each other and should therefore always be used as a set. To ensure that you have these correctly positioned the Basic loader at the end of this article is a repeat of last month's with the three new commands added.

CHANGE

Abbreviated entry: C(shift)H
Affected Basic abbreviations: CHR$ — CH (shift)R
Token: Hex $EE, $06 Decimal 238,6
Modes: Direct and program
Recommended mode: Direct only
Purpose: To change all occurrences of a string or command to something else.

CHANGE

```
1000 CHANGE JSR RESVAR ;RESET LINE LINKS
1010  JSR FIND14 ;GET CURRENT CHAR
1020  STA $59 ;STORE IN FLAG
1030  LDX #$00
1040  JSR FIND03 ;GET SEARCH STRING
1050  LDX #$00
1060  JSR CHAN07 ;GET STRING TO CHANGE
1070  STX $FC ;STORE LENGTH OF CHANGE STRING
1080  JSR FIND05 ;SETUP POINTERS
1090  SEI
1100  LDA $0300
1110  STA FINDER
1120  LDA $0301
1130  STA FINDER+1
1140  LDA #<FIND11 ;ERROR LINK TO RTS
1150  STA $0300
1160  LDA #>FIND11
1170  STA $0301
1180  CLI
1190  JSR FIND06 ;FIND STRING
1200 CHAN01 JMP CHAN03 ;CHANGE
1210 CHAN02 JSR FIND12 ;LIST LINE
1220  JSR FIND07 ;FIND STRING
1230  JMP CHAN01 ;AND REPEAT
1240 ;
1250 CHAN03 LDA $FC ;LENGTH OF CHANGE STRING
1260  SEC
1270  SBC $22 ;- LENGTH OF FIND
1280  BEQ CHAN04 ;THEY ARE EQUAL
1290  JMP CHAN10 ;ELSE CHANGE SIZE
1300 CHAN04 LDY $23 ;INDEX TO LINE
1310  LDX #$40 ;INDEX TO CHANGE STRING
1320  LDA $01
1330  AND #$FE ;OUT BASIC ROM
1340  STA $01
1350 CHAN05 LDA $BF40,X ;GET CHANGE CHAR
1360  BEQ CHAN06 ;END OF STRING
1370  STA ($57),Y ;REPLACE CHAR
1380  INX ;NEXT CHAR
1390  INY ;NEXT BYTE
1400  JMP CHAN05 ;AND AGAIN
1410 CHAN06 LDA $01
1420  ORA #$01 ;IN BASIC ROM
1430  STA $01
1440  DEY
1450  STY $23 ;STORE LINE INDEX
1460  JMP CHAN02 ;DO NEXT FIND
1470 ;
1480 CHAN07 JSR FIND13 ;GET NEXT CHAR
1490  CMP $59 ;IS IT THE FLAG?
1500  BEQ CHAN08 ;YES, GET STRING
1510  JMP $AF08
1520 CHAN08 JSR FIND13 ;GET NEXT CHAR
1530  BEQ CHAN09 ;END OF LINE
1540  CMP $59 ;END OF STRING?
1550  BEQ CHAN09 ;YES
1560  STA $BF80,X ;STORE CHAR
1570  INX
1580  CPX #$40 ;STRING TOO LONG?
1590  BNE CHAN08 ;NO
1600  LDX #$17 ;STRING TOO LONG
1610  JMP $A437 ;OUTPUT ERROR
1620 CHAN09 LDA #$00 ;STRING TERMINATOR
1630  STA $BF80,X ;STORE IT
1640  RTS
1650 ;
1660 CHAN10 LDY #$00
1670  LDA ($57),Y ;GET LINE# LO
1680  STA $14 ;STORE IT
1690  INY
1700  LDA ($57),Y ;GET LINE# HI
1710  STA $15 ;STORE IT
1720  LDX #$00
1730 CHAN11 INY
1740  CPY $23 ;REACHED STRING?
1750  BEQ CHAN12 ;YES, INSERT IT
1760  LDA ($57),Y ;GET PROGRAM BYTE
1770  STA $0200,X ;STORE IN BUFFER
1780  INX
1790  CPX #$56 ;BUFFER TOO LARGE?
1800  BNE CHAN11 ;NOT YET
1810 CHAN12 LDA $01
1820  AND #$FE ;OUT BASIC ROM
1830  STA $01
1840  LDY #$00
1850 CHAN13 LDA $BF80,Y ;GET CHANGE STRING BYTE
1860  BEQ CHAN14 ;END OF STRING
1870  STA $0200,X ;STORE IN BUFFER
1880  INX ;NEXT CHAR
1890  INY ;AND PROGRAM BYTE
1900  CPX #$57 ;END OF BUFFER?
1910  BNE CHAN13 ;NO
1920 CHAN14 LDA $01
1930  ORA #$01 ;IN BASIC ROM
```

```
1940  STA $01
1950  LDA $23 ;CALCULATE START
1960  CLC ;OF REST OF PROGRAM LINE
1970  ADC $22 ;AFTER INSERTING THE
1980  TAY ;CHANGE STRING
1990  LDA $23
2000  CLC
2010  ADC $FC
2020  STA $23
2030  DEC $23
2040 CHAN15 LDA ($57),Y ;GET PROGRAM BYTE
2050  STA $0200,X ;STORE IN BUFFER
2060  INY ;NEXT BYTE
2070  INX ;NEXT CHAR
2080  CMP #$00 ;END OF LINE?
2090  BEQ CHAN16 ;YES
2100  CPX #$58 ;END OF BUFFER?
2110  BNE CHAN15 ;NOT YET
2120  LDA #$00 ;ZERO IF END OF BUFFER
2130  STA $0200,X ;STORE IT
2140  INX
2150 CHAN16 STX CHANLN ;STORE LENGTH OF
2160  TXA ;LINE
2170  CLC
2180  ADC #$04
2190  STA $0B
2200  LDA $0302
2210  STA CHANST
2220  LDA $0303
2230  STA CHANST+1
2240  LDA #<CHAN17 ;BASIC WARM START
2250  STA $0302 ;RE-ENTRY POINT
2260  LDA #>CHAN17
2270  STA $0303
2280  JSR FIND15 ;SAVE POINTERS ETC
2290  LDY $0B ;GET POINTER
2300  JMP $A4A4 ;INSERT PROGRAM LINE
2310 CHAN17 LDA CHANST ;RESTORE WARM START VECTOR
2320  STA $0302
2330  LDA CHANST+1
2340  STA $0303
2350  JSR FIND16 ;RESTORE POINTERS ETC
2360  LDA $57 ;LAST LINE?
2370  CMP $2D
2380  BNE CHAN18 ;NOT YET
2390  LDA $58
2400  CMP $2E
2410  BEQ CHAN20 ;YES
2420 CHAN18 LDA CHANLN ;DID WE DELETE
2430  CMP #$01 ;WHOLE LINE?
2440  BEQ CHAN19 ;YES
2450  JMP CHAN02 ;NO, LIST AND DO NEXT
2460 CHAN19 LDY #$02 ;INDEX TO NEXT LINE
2470  STY $23
2480  LDX #$00
2490  JMP CHAN02+3 ;DO NEXT WITHOUT LIST
2500 CHAN20 JMP FIND10 ;EXIT CHANGE
2510 CHANLN .BYT 0
2520 CHANST .WOR 0
2530 .END
```

DUMP

```
1000 DUMP LDA $2E ;GET START OF VARIABLES
1010  STA $60 ; AND STORE IN REQUIRED
1020  LDA $2D ; LOCATIONS
1030  STA $5F
1040 ;
1050 DUMP01 SEC ;START OF MAIN LOOP
1060  SBC $2F ;END OF VARIABLES?
1070  LDA $60
1080  SBC $30
1090  BCC DUMP02 ;NO
1100  JMP DUMP17 ;YES, DISPLAY ARRAY DIMS
1110 ;
1120 DUMP02 JSR DUMP12 ;GET VAR NAME
1130  LDA $25 ;REAL?
1140  BEQ DUMP03 ;YES
1150  CMP #$01 ;FUNCTION?
1160  BEQ DUMP26 ;YES
1170  CMP #$02 ;STRING?
1180  BEQ DUMP04 ;YES
1190  LDA #$25 ;MUST BE INTEGER
1200  JSR $FFD2 ;PRINT '%'
1210  JSR DUMP15 ;ANY EXTRA SPACES
1220  LDA #$3D
1230  JSR $FFD2 ;PRINT '='
1240  LDY #$02 ;SET POINTER TO VAL
1250  LDA ($5F),Y ;GET LO
1260  PHA
1270  INY
1280  LDA ($5F),Y ;GET HI
1290  TAY
1300  PLA
1310  JSR $B391 ;FIX-FLOAT
1320  JSR $BDDD ;FLOAT-ASCII
1330  JSR $AB1E ;PRINT IT
1340  JMP DUMP07 ;DO NEXT VAR
1350 ;
1360 ;REAL VARIABLE
1370 ;
1380 DUMP03 LDA #$20
1390  JSR $FFD2 ;PRINT SPACE
1400  JSR DUMP15 ;PAD NAME
1410  LDA #$3D
1420  JSR $FFD2 ;PRINT '='
1430  JSR $B185 ;GET ADDRESS OF VAR
1440  LDA $47 ; INTO A AND Y
1450  LDY $48
1460  JSR $BBA2 ;MEM-FAC#1
1470  JSR $BDDD ;FLOAT-ASCII
```

Each line that is changed is listed if there is anything left to list.

Syntax: CHANGE dstr1 ddstr2d – where d is a delimiter character that does not appear in either of the strings (str1 or str2).

Errors: Syntax error – if the format is not as above. String too long – if either str1 or str2 are longer than 40 characters.

Use: CHANGE has a number of uses. An example would be:

CHANGE PRINT PRINT#4,

To change all occurrences of 'PRINT' to 'PRINT#4,', or:

CHANGE "PRINT""PRINT#4,"

which will change all occurrences of the text 'PRINT' to the text 'PRINT#4,'.

Note: Not all delimeter characters will work in all cases, for example:

CHANGE /REM///

As the character '/' has two values the first is the token for divide and the second is just the ASCII slash character.

The same is true of DATA. Other characters that will have the same effect are: '+-* = '.

Routine entry point: $86BB

Routine operation: CHANGE uses most of the FIND routines to find str1 and list the line.

CHANGE reads in the delimiter byte and stores it away. The string to be changed is then read in until the second delimiter character is reached and stored. The next character is checked to see that it equals the delimiter character and if so the string to change to is read in until the delimiter character is found again or the end of command.

The rest of the routine is just a loop finding all occurrences, changing them and listing until the end of the program.

The actual routine that changes the string uses the Basic input buffer and the Basic routines to change a line. The routine copies the line up to str1 into the buffer, the change string (str2) is then copied to the buffer and the remainder of the line is copied over. The pointers are then set so that the next byte to check is the one following str2.

DUMP

Abbreviated entry: D(shift)U
Affected Basic abbreviations: None
Token: Hex $EE, $0C Decimal 238,12
Modes: Direct and program
Recommended mode: Direct
Purpose: To display the values of all simple variations, name functions, and display the dimensions of arrays.
Syntax: DUMP
Errors: None
Use: For de-bugging Basic programs, the DUMP command may be used after the program has run to get a list of all variables and their values. As an added bonus, not found in any other DUMP command for the Commodore 64, all array dimensions are also given. The DUMP command will also display function names.
Routine entry point: $8B02
Routine operation: The DUMP routine sets a pointer to the start of variables and checks for the end of variables. If it does not find any, the variable name is read in and displayed, the variable type is determined, and the display is produced according to which type is required. When all simple variables have been processed, arrays are handled. The array names are read and displayed in the same way as the simple variables and the number of dimensions read off. The pointer is then set to the end of the dimension entries and, reading backwards, the dimensions are read and displayed.

FIND

Abbreviated entry: F(shift)I
Affected Basic abbreviations: None
Token: Hex $EE,$0E Decimal 238,14
Modes: Direct and program
Recommended mode: Direct only
Purpose: To find all occurrences of a string or command inside a Basic program.
Syntax: FIND string – where d is the delimiter character as in CHANGE.
Errors: Syntax error – if the syntax is not as above. String too long – if the string is longer than 40 characters.
Use: FIND is another useful routine for de-bugging and

DUMP

```
1480  JSR $BDDA ;PRINT NUMBER
1490  JMP DUMP07 ;DO NEXT VAR
1500 :
1510 ;FUNCTION
1520 :
1530 DUMP26 JSR DUMP15 ;PAD NAME
1540  LDA #<FUNCTT ;POINT TO
1550  LDY #>FUNCTT ;'FUNCTION'
1560  JSR $AB1E ;PRINT STRING
1570  JMP DUMP07 ;DO NEXT VAR
1580 FUNCTT .BYT ' = FUNCTION',$00
1590 :
1600 ;STRING VARIABLE
1610 :
1620 DUMP04 LDX #$03 ;LOOP TO PRINT '$= "'
1630 DUMP05 LDA DUMTBL,X
1640  JSR $FFD2
1650  CPX #$03
1660  BNE DUMP06
1670  JSR DUMP15 ;PAD FOR NAME
1680 DUMP06 DEX
1690  BPL DUMP05 ;COMPLETE LOOP
1700  LDY #$04 ;GET ADDRESS OF STRING
1710  LDA ($5F),Y
1720  STA $23
1730  DEY
1740  LDA ($5F),Y
1750  STA $22
1760  DEY
1770  LDA ($5F),Y ;LENGTH
1780  JSR $AB24 ;PRINT STRING FROM ($22)
1790  LDA #$22 ; AND LENGTH IN .A
1800  JSR $FFD2 ;PRINT '"'
1810 :
1820 ;PRINT CARRIAGE RETURN AND DO NEXT
1830 :
1840 DUMP07 LDA #$0D
1850  JSR $FFD2 ;PRINT RETURN
1860 DUMP08 JSR $FFE1 ;STOP KEY?
1870  BNE DUMP10 ;NO
1880 DUMP09 RTS ;EXIT TO 'READY'
1890 DUMP10 CLC ;MOVE TO NEXT VAR
1900  LDA $5F
1910  ADC #$07
1920  STA $5F
1930  LDX $60
1940  BCC DUMP11
1950  INX
1960 DUMP11 STX $60
1970  JMP DUMP01 ;DO NEXT VAR
1980 :
1990 ;GET AND PRINT VAR NAME
2000 :
2010 DUMP12 LDY #$00 ;GET VARIABLE TYPE
2020  STY $25  ;AND NAME
2030  INY
2040 DUMP13 LDA ($5F),Y ;GET BYTE
2050  ASL A ;TYPE BIT INTO TEMP
2060  ROL $25
2070  LSR A ;RESTORE NAME BYTE
2080  STA $0045,Y ;STORE NAME BYTE
2090  DEY
2100  BPL DUMP13
2110  LDA $45 ;PRINT NAME
2120  JSR $FFD2
2130  LDA $46 ;2ND BYTE?
2140  BEQ DUMP14 ;NO
2150  JSR $FFD2 ;YES, PRINT IT
2160 DUMP14 RTS ;DONE
2170 :
2180 ;PAD OUT NAME IF ONLY 1 BYTE LONG
2190 :
2200 DUMP15 LDA $46 ;2ND BYTE?
2210  BNE DUMP16 ;YES, DON'T PAD
2220  LDA #$20 ;ELSE PAD WITH SPACE
2230  JSR $FFD2 ;PRINT
2240 DUMP16 RTS ;DONE
2250 :
2260 ;DISPLAY ARRAY NAMES AND DIMENSIONS
2270 ;ONLY
2280 :
2290 DUMP17 LDA #$0D ;SEPARATE NORMAL
2300  JSR $FFD2 ; VARS FROM ARRAYS WITH
2310  LDA $2F ; A CARRIAGE RETURN
2320  STA $5F ;SET POINTER TO 1ST
2330  LDA $30 ; ARRAY
2340  STA $60
2350 :
2360 DUMP18 LDA $60 ;END OF ARRAYS?
2370  CMP $32
2380  BNE DUMP19 ;NO
2390  LDA $5F
2400  CMP $31
2410  BEQ DUMP09
2420 :
2430 DUMP19 JSR $FFE1 ;STOP KEY?
2440  BEQ DUMP09 ;YES,EXIT
2450  JSR DUMP12 ;GET AND PRINT NAME
2460  LDA $25 ;WHICH TYPE?
2470  BEQ DUMP21 ;REAL
2480  CMP #$02 ;STRING?
2490  BNE DUMP20 ;NO, ARRAY IS INTEGER
2500  LDA #$24 ;CHAR '$'
2510  .BYT $2C ;SKIP 2 BYTES
2520 DUMP20 LDA #$25 ;CHAR '%'
2530  .BYT $2C ;SKIP 2 BYTES
2540 DUMP21 LDA #$20 ;CHAR ' '
2550  JSR $FFD2 ;PRINT IT
2560  JSR DUMP15
2570  LDA #$20
2580  JSR $FFD2 ;ONE EXTRA SPACE
2590  LDA #$28 ;CHAR '('
2600  JSR $FFD2 ;PRINT IT
2610  LDA $5F
```

DUMP

```
2620  CLC
2630  ADC #$03 ;SET POINTER TO END
2640  STA $FB : OF ARRAY ENTRY FOR
2650  LDA $60 ; DISPLAY OF DIMS
2660  ADC #$00
2670  STA $FC
2680  LDY #$01
2690  LDA ($FB),Y ;# OF DIMENSIONS
2700  STA $FD
2710  LDA #$00
2720  STA $FE
2730  ASL $FD ;TIMES 2
2740  ROL $FE
2750  LDA $FD ;PLUS END VALUE
2760  CLC
2770  ADC $FB
2780  STA $FD
2790  LDA $FE
2800  ADC $FC
2810  STA $FE
2820 :
2830 DUMP22 LDY #$00 ;GET DIMENSION VALUE
2840  LDA ($FD),Y
2850  STA DIMENS+1
2860  INY
2870  LDA ($FD),Y
2880  STA DIMENS
2890  BNE DUMP23 ;MINUS 1
2900  DEC DIMENS+1
2910 DUMP23 DEC DIMENS
2920  LDA DIMENS+1 ;PRINT NUMBER
2930  LDX DIMENS : IN .A(HI), .X(LO)
2940  LDY $5F ;SAVE ARRAY POINTER
2950  STY DIMENS
2960  LDY $60
2970  STY DIMENS+1
2980  JSR $BDCD
2990  LDY DIMENS ;RESTORE ARRAY POINTER
3000  STY $5F
3010  LDY DIMENS+1
3020  STY $60
3030  SEC :SUBTRACT 2 FROM
3040  LDA $FD : DIMENSION POINTER
3050  SBC #$02
3060  STA $FD
3070  LDA $FE
3080  SBC #$00
3090  STA $FE
3100  CMP $FC :END OF ARRAY?
3110  BNE DUMP24 ;NO
3120  LDA $FD
3130  CMP $FB
3140  BEQ DUMP25 ;YES
3150 :
3160 DUMP24 LDA #$2C ;CHAR ','
3170  JSR $FFD2 :PRINT IT
3180  JMP DUMP22 :DO NEXT ELEMENT
3190 :
3200 DUMP25  LDY #$03 ;GET LENGTH OF
3210  LDA ($5F),Y ;ARRAY ENTRY
3220  STA $FB
3230  DEY
3240  LDA ($5F),Y
3250  CLC
3260  ADC $5F ;AND ADD TO ARRAY
3270  STA $5F ; POINTER
3280  LDA $60
3290  ADC $FB
3300  STA $60
3310  LDA #$29 ;CHAR ')'
3320  JSR $FFD2 ;PRINT IT
3330  LDA #$0D ;CARRIAGE RETURN
3340  JSR $FFD2 ;PRINT IT
3350  JMP DUMP18 ;DO NEXT ARRAY
3360 DIMENS .WOR 0
3370 DUMTBL .BYT $22,$20,$3D,$24
3380 .END
```

FIND

```
1000 FIND JSR FIND14 ;GET CHARACTER
1010  STA $59 ;STORE IN FLAG
1020  LDX #$00
1030  JSR FIND03 ;GET SEARCH STRING
1040  JSR FIND05 ;SETUP POINTERS
1050  SEI
1060  LDA $0300
1070  STA FINDER
1080  LDA $0301
1090  STA FINDER+1
1100  LDA #<FIND11 :ERROR LINK TO RTS
1110  STA $0300
1120  LDA #>FIND11
1130  STA $0301
1140  CLI
1150  JSR FIND06 ;FIND STRING
1160 FIND01 JSR FIND12 ;LIST LINE
1170  JSR FIND07 ;FIND STRING
1180  JMP FIND01 ;AND REPEAT
1190 ;
1200 FIND02 JMP $AF08 ;SEND SYNTAX ERROR
1210 :
1220 FIND03 JSR FIND13 ;GET A CHARACTER
1230  BEQ FIND02 ;END OF LINE
1240  CMP $59 ;END OF STRING?
1250  BEQ FIND04 ;YES, COMPLETE
```

checking Basic programs, for example:

FIND PRINT

which will find and list all lines containing the command PRINT. If PRINT occurs more than once on a line, the line will be listed each time it is found with the exception of the last lines where the line will be listed only once.

Routine entry point: $8D93

Routine operation: The string to be found is read in within quotes, including spaces and colons and stored away. The rest of the program is a loop that searches the program until the string has been found, lists the line, and starts searching from the next character.

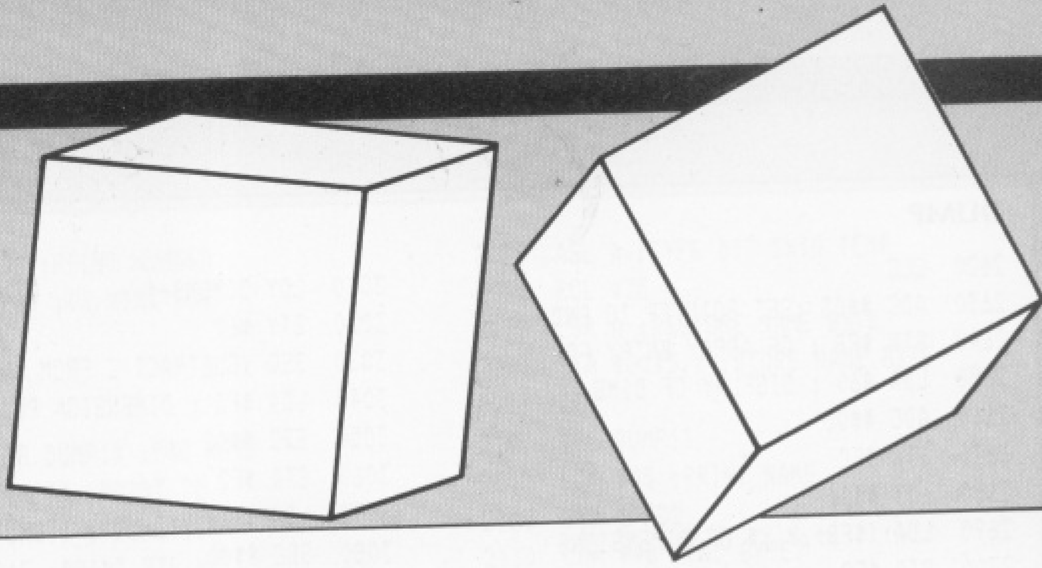
The error message vector is stored away and replaced with a jump to an 'RTS' so that LIST will return to the routine.

APPEND

Abbreviated entry: A(shift)P
Affected Basic abbreviations: NONE
Token: Hex $EE,$03 Decimal 238,3
Modes: Direct and program
Recommended mode: Direct
Purpose: To load a program into memory so that it appears 'on top' of the current program. This routine will work with both disk and cassette and the variable pointers when loaded are set to the end of the combined program. When this routine is used, you should check that the line numbers of the APPENDed program are larger than the line numbers of the program in memory.
Syntax: APPEND [filename[,d[,s]]] – where d is the device number and s is the secondary address.
Errors: The same errors will be encountered as in the Basic command LOAD.
Use: This routine would be used mostly to add Basic library routines onto the end of your programs. It would be used rather than MERGE because APPEND is much faster.
Routine entry point: $84D8
Routine operation: The APPEND routine uses LOAD's parameter parsing routine to get the filename etc. then sets the secondary address so that it loads at the end of the Basic program in memory. The load

routine is then called and the program re-chained and variable pointers set.

These extended Basic routines are all taken from the book *Advanced Commodore 64 Basic Revealed* by Nick Hampshire, published by Collins.



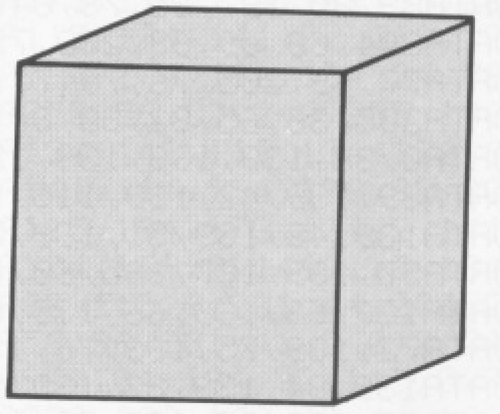
FIND

```
1260  STA $BF40,X ;STORE IN SEARCH STRING
1270  INX
1280  CPX #$40 ;STRING TOO LONG?
1290  BNE FIND03 ;NO
1300  LDX #$17 ;STRING TOO LONG
1310  JMP $A437 ;OUTPUT ERROR
1320 FIND04 LDA #$00 ;TERMINATOR TO STRING
1330  STA $BF40,X ;STORE IT
1340  STX $22 ;STORE STRING LENGTH
1350  RTS ;EXIT
1360 ;
1370 FIND05 LDA $2B ;GET START OF PROGRAM
1380  CLC
1390  ADC #$02 ;PLUS 2
1400  STA $57
1410  LDA $2C ;GET START OF PROG MSB
1420  ADC #$00
1430  STA $58 ;STORE IT
1440  RTS
1450 ;
1460 FIND06 LDX #$00 ;INDEX TO STRING
1470  LDY #$02 ;INDEX TO LINE
1480  STY $23
1490 FIND07 LDA $01
1500  AND #$FE ;OUT BASIC ROM
1510  STA $01
1520  LDA ($57),Y ;GET BYTE
1530  BEQ FIND09 ;END OF LINE
1540  CMP $BF40,X ;SAME AS STRING?
1550  PHP
1560  LDA $01
1570  ORA #$01 ;IN BASIC ROM
1580  STA $01
1590  PLP
1600  BNE FIND08 ;NOT MATCHED
1610  INY ;NEXT BYTE
1620  INX ;NEXT CHAR
1630  CPX $22 ;STRING MATCHED?
1640  BNE FIND07 ;NO
1650  RTS ;YES
1660 FIND08 INC $23 ;START AT NEXT BYTE
1670  LDY $23
1680  LDX #$00 ;AND START OF STRING
1690  LDA ($57),Y ;GET BYTE
1700  BEQ FIND09 ;END OF LINE
1710  JMP FIND07 ;TRY AGAIN
1720 FIND09 LDA $01
1730  ORA #$01 ;IN BASIC ROM
1740  STA $01
1750  LDA $57
1760  SEC
1770  SBC #$02 ;LINE POINTER -2
1780  STA $57
1790  LDA $58
1800  SBC #$00
1810  STA $58
1820  LDY #$00
1830  LDA ($57),Y ;GET LINK LO
1840  STA $59 ;STORE IT
1850  INY
1860  LDA ($57),Y ;GET LINK HI
1870  STA $58 ;STORE TO POINTER HI
1880  ORA $59 ;END OF PROGRAM?
1890  BEQ FIND10 ;YES
1900  LDA $59 ;GET LINE POINTER LO
1910  CLC
1920  ADC #$02 ;ADD 2
1930  STA $57 ;STORE IT
1940  LDA $58 ;GET HI BYTE
1950  ADC #$00
1960  STA $58
1970  JMP FIND06 ;DO NEXT LINE
1980 FIND10 SEI
1990  LDA FINDER ;RESET ERROR LINK
2000  STA $0300
2010  LDA FINDER+1
2020  STA $0301
2030  CLI
2040  JMP $A474 ;EXIT
2050 ;
2060 FIND11 RTS ;ERROR LINK
2070 ;
2080 FIND12 LDY #$00
2090  JSR FIND15 ;SAVE POINTERS
2100  LDA #$91 ;CURSOR UP
2110  JSR $FFD2 ;PRINT IT
2120  LDA ($57),Y ;GET LINE# LO
2130  STA $14 ;STORE IT
2140  INY
2150  LDA ($57),Y ;GET LINE# HI
2160  STA $15 ;STORE IT
2170  JSR $A613 ;FIND LINE ADDRESS
2180  JSR $A6C9 ;LIST LINE
2190  JSR FIND16 ;RESTORE POINTERS
2200  INC $23 ;NEXT CHAR IN LINE
2210  LDY $23
2220  LDX #$00 ;START OF STRING
2230  RTS
2240 ;
2250 FIND13 INC $7A ;INCREASE LSB
2260  BNE FIND14
2270  INC $7B
2280 FIND14 LDY #$00
2290  LDA ($7A),Y ;GET INPUT BYTE
2300  RTS
2310 ;
2320 FIND15 LDA $22 ;STORE STRING LENGTH
2330  STA FIND17
2340  LDA $23 ;STORE LINE INDEX
2350  STA FIND17+1
2360  LDA $57 ;STORE LINE POINTER LO
2370  STA FIND17+2
2380  LDA $58 ;HI
2390  STA FIND17+3
2400  LDA $FC ;SAVE CHANGE VARIABLE
2410  STA FIND17+4
2420  RTS
2430 ;
2440 FIND16 LDA FIND17 ;GET STRING LENGTH
2450  STA $22
2460  LDA FIND17+1 ;GET LINE INDEX
2470  STA $23
2480  LDA FIND17+2 ;GET LINE POINTER LO
2490  STA $57
2500  LDA FIND17+3 ;GET LINE POINTER HI
2510  STA $58
2520  LDA FIND17+4 ;GET CHANGE PARAMETER
2530  STA $FC
2540  RTS
2550 FIND17 .BYT $00,$00,$00,$00,$00
2560 FINDER .WOR 0
2570 .END
```

APPEND

```
1000 APPEND LDA #$00
1010   STA $0A
1020   JSR $E1D4 ; GET FILE PARAMETERS
1030   LDA #$00
1040   STA $B9 ;SET SA FOR ALT LOAD
1050   LDA $2D
1060   SEC
1070   SBC #$02 ; SET LOAD ADDRESS
1080   TAX ; DIRECTLY AFTER RESIDENT
1090   LDA $2D+1 ; PROGRAM.
1100   SBC #$00
1110   TAY
1120   LDA $0A
1130   JSR $FFD5 ; LOAD
1140 ;
1150 RESVAR JSR $A533 ;RE-CHAIN LINES
1160   LDA $2D
1170   LDY $2D+1 ; RESET VARIABLE
1180   SEC ; POINTERS TO END OF
1190   SBC #$02 ; NEW PROGRAM
1200   STA $57
1210   TYA
1220   SBC #$00
1230   STA $57+1
1240 RESV01 LDY #$00 ; FIND END OF PROGRAM
1250   LDA ($57),Y ; AND SET VARIABLE
1260   BNE RESV02 ; POINTERS
1270   INY
1280   LDA ($57),Y
1290   BNE RESV02
1300   LDA $57
1310   CLC
1320   ADC #$02
1330   STA $2D
1340   STA $2F
1350   STA $31
1360   LDA $57+1
1370   ADC #$00
1380   STA $2D+1
1390   STA $2F+1
1400   STA $31+1
1410   RTS
1420 RESV02 LDY #$00 ; NOT YET END OF
1430   LDA ($57),Y ; PROGRAM. GET
1440   STA $59 ; ADDRESS OF NEXT
1450   INY ; LINE.
1460   LDA ($57),Y
1470   STA $57+1
1480   LDA $59
1490   STA $57
1500   JMP RESV01
1510 .END
```

BASIC LOADER



```
100 REM ********************************
110 REM *BASIC LOADER FOR THE BASIC    *
120 REM *EXTENSION PACKAGE             *
130 REM *INCLUDES WEDGES AND THE       *
140 REM *COMMANDS:                     *
150 REM *APPEND, CHANGE, CTL, DUMP AND*
160 REM *FIND                          *
170 REM *COPYRIGHT 20.8.85             *
180 REM *NICK HAMPSHIRE                *
190 REM ********************************
200 I=1:X=0:L=32768
210 READA:IFA=999THEN300
220 POKEL,A
230 L=L+1:I=I+1:X=X+A
240 GOTO210
300 IFI<>2927THENPRINT"NUMBER OF
DATA ENTRIES ERROR "I" SHOULD
BE 2927":END
310 IFX<>341614 THENPRINT"CHECKSUM
ERROR. VALUE "X" SHOULD BE
341614":END
320 REM TO RUN ROUTINES SYS(64738)
330 REM
340 END
1000 DATA122,128,57,128,195,194,205
1010 DATA56,48,139,227,131,164,201
1020 DATA129,158,130,247,130,59,131
1030 DATA76,72,178,0,49,234,68
1040 DATA128,71,254,74,243,145,242
1050 DATA14,242,80,242,51,243,241
1060 DATA131,202,241,237,246,62,241
1070 DATA47,243,68,128,165,244,237
1080 DATA245,32,188,246,32,225,255
1090 DATA240,3,76,114,254,32,163
1100 DATA253,32,24,229,32,93,128
1110 DATA32,204,255,169,0,133,19
1120 DATA32,122,166,88,162,128,76
1130 DATA136,227,162,21,160,128,134
1140 DATA195,132,196,160,35,177,195
1150 DATA153,16,3,136,16,248,169
1160 DATA118,160,131,141,143,2,140
1170 DATA144,2,96,142,22,208,32
1180 DATA163,253,32,80,253,32,91
1190 DATA255,32,93,128,88,32,229
1200 DATA128,32,191,227,169,128,133
1210 DATA52,133,54,133,56,169,0
1220 DATA133,51,133,53,133,55,169
1230 DATA172,160,128,32,45,228,162
1240 DATA251,154,208,172,147,13,32
1250 DATA32,32,32,42,42,42,42
1260 DATA32,69,88,84,69,78,68
1270 DATA69,68,32,54,52,32,66
1280 DATA65,83,73,67,32,86,48
1290 DATA49,32,42,42,42,42,13
1300 DATA13,32,54,52,75,32,82
1310 DATA65,77,32,83,89,83,84
```

BASIC LOADER

```
1320 DATA69,77,32,32,0,162,11
1330 DATA189,9,128,157,0,3,202
1340 DATA16,247,96,82,85,206,67
1350 DATA84,204,65,80,80,69,78
1360 DATA196,65,85,84,207,67,65
1370 DATA84,65,76,79,199,67,72
1380 DATA65,78,71,197,67,72,65
1390 DATA73,206,67,82,85,78,67
1400 DATA200,68,69,76,69,84,197
1410 DATA68,73,83,203,68,79,75
1420 DATA197,68,85,77,208,69,88
1430 DATA69,195,70,73,78,196,71
1440 DATA69,212,75,69,217,77,65
1450 DATA212,77,69,82,71,197,79
1460 DATA76,196,80,79,208,80,85
1470 DATA212,82,69,78,85,77,66
1480 DATA69,210,82,69,80,69,65
1490 DATA212,83,79,82,212,84,82
1500 DATA65,67,69,79,206,84,82
1510 DATA65,67,69,79,70,198,84
1520 DATA89,80,197,85,78,84,73
1530 DATA204,68,69,69,203,72,73
1540 DATA77,69,205,76,79,77,69
1550 DATA205,86,65,82,80,84,210
1560 DATA0,36,139,138,133,197,138
1570 DATA39,139,42,139,116,134,45
1580 DATA139,48,139,51,139,54,139
1590 DATA57,139,187,135,60,139,135
1600 DATA137,63,139,66,139,69,139
1610 DATA72,139,75,139,78,139,81
1620 DATA139,84,139,87,139,90,139
1630 DATA93,139,96,139,99,139,102
1640 DATA139,105,139,108,139,111,139
1650 DATA114,139,166,122,160,4,132
1660 DATA15,189,0,2,16,7,201
1670 DATA255,240,43,232,208,244,201
1680 DATA32,240,36,133,8,201,34
1690 DATA240,71,36,15,112,26,201
1700 DATA63,208,4,169,153,208,18
1710 DATA201,48,144,4,201,60,144
1720 DATA10,76,70,130,169,238,44
1730 DATA5,11,164,113,232,200,153
1740 DATA251,1,201,238,240,49,185
1750 DATA251,1,240,34,56,233,58
1760 DATA240,4,201,73,208,2,133
1770 DATA15,56,233,85,208,174,133
1780 DATA8,189,0,2,240,219,197
1790 DATA8,240,215,200,153,251,1
1800 DATA232,208,240,153,253,1,198
1810 DATA123,169,255,133,122,96,165
1820 DATA11,200,153,251,1,76,207
1830 DATA129,132,113,160,255,134,122
1840 DATA202,169,1,133,11,200,232
1850 DATA189,0,2,56,249,241,128
1860 DATA240,245,201,128,240,156,166
1870 DATA122,230,11,200,185,240,128
1880 DATA16,250,185,241,128,208,228
1890 DATA160,0,132,11,136,166,122
1900 DATA202,200,232,189,0,2,56
1910 DATA249,158,160,240,245,201,128
1920 DATA208,3,76,255,129,166,122
1930 DATA230,11,200,185,157,160,16
1940 DATA250,185,158,160,208,225,189
1950 DATA0,2,76,1,130,48,3
1960 DATA76,243,166,201,255,240,249
1970 DATA36,15,48,245,201,238,240
1980 DATA5,32,217,130,48,3,32
1990 DATA186,130,76,239,166,200,177
2000 DATA95,170,132,73,160,255,202
2001 =0:X=0
2010 DATA240,8,200,185,241,128,16
2020 DATA250,48,245,200,185,241,128
2030 DATA48,5,32,210,255,208,245
2040 DATA96,56,233,127,170,132,73
2050 DATA160,255,202,240,8,200,185
2060 DATA158,160,16,250,48,245,200
2070 DATA185,158,160,48,230,32,210
2080 DATA255,208,245,32,115,0,201
2090 DATA238,240,10,201,153,240,38
2100 DATA32,121,0,76,231,167,32
2110 DATA14,131,76,174,167,230,122
2120 DATA208,2,230,123,160,0,177
2130 DATA122,56,233,1,10,168,185
2140 DATA138,129,72,185,137,129,72
2150 DATA76,115,0,32,46,131,76
2160 DATA174,167,173,58,131,72,173
2170 DATA57,131,72,76,115,0,235
2180 DATA132,169,0,133,13,32,115
2190 DATA0,201,238,240,6,32,121
2200 DATA0,76,141,174,230,122,208
2210 DATA2,230,123,160,0,177,122
2220 DATA201,29,176,3,76,8,175
2230 DATA133,36,169,173,72,169,140
2240 DATA72,198,36,165,36,10,170
2250 DATA189,138,129,72,189,137,129
2260 DATA72,76,115,0,165,157,240
2270 DATA16,169,1,36,212,208,10
2280 DATA165,203,201,3,144,4,201
2290 DATA7,144,3,76,72,235,197
2300 DATA197,240,249,169,0,133,252
2310 DATA133,251,169,1,44,141,2
2320 DATA240,4,169,32,133,251,169
2330 DATA191,133,252,169,192,24,101
2340 DATA251,133,251,165,203,201,3
2350 DATA208,4,169,24,208,18,201
2360 DATA6,208,4,169,16,208,10
2370 DATA201,5,208,4,169,8,208
2380 DATA2,169,0,24,101,251,133
2390 DATA251,160,0,169,54,133,1
2400 DATA177,251,240,8,153,119,2
2410 DATA200,192,8,208,244,132,198
```

BASIC LOADER

```
2420 DATA169,55,133,1,165,203,133
2430 DATA197,173,141,2,141,142,2
2440 DATA96,165,153,208,4,165,157
2450 DATA208,3,76,87,241,165,211
2460 DATA133,202,165,214,133,201,152
2470 DATA72,138,72,165,208,240,6
2480 DATA76,58,230,32,22,231,165
2490 DATA198,133,204,141,146,2,240
2500 DATA247,120,165,207,240,12,165
2510 DATA206,174,135,2,160,0,132
2520 DATA207,32,19,234,32,180,229
2530 DATA201,131,208,16,162,9,120
2540 DATA134,198,189,230,236,157,118
2550 DATA2,202,208,247,240,207,201
2560 DATA13,208,3,76,2,230,201
2570 DATA17,208,193,166,214,224,24
2580 DATA240,3,76,15,132,162,24
2590 DATA160,0,24,32,240,255,230
2600 DATA20,208,2,230,21,32,19
2610 DATA166,160,1,177,95,208,16
2620 DATA169,255,133,20,133,21,169
2630 DATA185,160,132,32,30,171,76
2640 DATA18,132,160,2,177,95,133
2650 DATA20,200,177,95,133,21,169
2660 DATA162,141,0,3,169,132,141
2670 DATA1,3,104,141,183,132,104
2680 DATA141,184,132,160,1,132,15
2690 DATA76,215,166,169,139,141,0
2700 DATA3,169,227,141,1,3,173
2710 DATA184,132,72,173,183,132,72
2720 DATA76,18,132,0,1,13,13
2730 DATA18,42,42,42,42,42,42
2740 DATA42,42,42,42,42,42,32
2750 DATA69,78,68,32,79,70,32
2760 DATA80,82,79,71,82,65,77
2770 DATA32,42,42,42,42,42,42
2780 DATA42,42,42,42,42,42,13
2790 DATA0,32,33,171,32,121,0
2800 DATA240,80,240,94,201,163,240
2810 DATA107,201,166,24,240,102,201
2820 DATA238,208,20,160,1,177,122
2830 DATA201,2,208,12,32,115,0
2840 DATA32,115,0,32,139,133,76
2850 DATA233,132,32,121,0,201,44
2860 DATA240,55,201,59,240,97,32
2870 DATA158,173,36,13,48,195,32
2880 DATA221,189,32,135,180,32,33
2890 DATA171,32,59,171,208,184,169
2900 DATA0,157,0,2,162,255,160
2910 DATA1,165,19,208,16,169,13
2920 DATA32,71,171,36,19,16,5
2930 DATA169,10,32,71,171,73,255
2940 DATA96,56,32,240,255,152,56
2950 DATA233,10,176,252,73,255,105
2960 DATA1,208,25,8,56,32,240
2970 DATA255,132,9,32,155,183,201
2980 DATA41,240,3,76,8,175,40
2990 DATA144,6,138,229,9,144,5
3000 DATA170,232,202,208,6,32,115
3010 DATA0,76,238,132,32,59,171
3020 DATA208,242,76,30,171,32,12
3030 DATA134,32,250,174,32,121,0
3040 DATA32,49,134,176,8,32,69
3050 DATA134,142,111,134,176,66,32
3060 DATA46,134,176,8,32,78,134
3070 DATA142,112,134,176,53,32,46
3080 DATA134,176,8,32,72,134,142
3090 DATA113,134,176,40,32,46,134
3100 DATA176,8,32,72,134,142,114
3110 DATA134,176,27,32,46,134,176
3120 DATA8,32,72,134,142,115,134
3130 DATA176,14,32,46,134,144,3
3140 DATA76,8,175,32,75,134,142
3150 DATA116,134,32,247,174,173,116
3160 DATA134,240,5,169,147,32,22
3170 DATA231,173,113,134,141,134,2
3180 DATA173,114,134,141,33,208,173
3190 DATA115,134,141,32,208,172,111
3200 DATA134,174,112,134,24,76,240
3210 DATA255,56,32,240,255,140,111
3220 DATA134,142,112,134,173,33,208
3230 DATA141,114,134,173,32,208,141
3240 DATA115,134,173,134,2,141,113
3250 DATA134,169,0,141,116,134,96
3260 DATA32,115,0,201,44,208,2
3270 DATA56,96,201,41,240,2,24
3280 DATA96,104,104,32,115,0,76
3290 DATA230,133,169,40,44,169,16
3300 DATA44,169,2,44,169,25,141
3310 DATA110,134,32,158,183,236,110
3320 DATA134,176,14,32,121,0,201
3330 DATA41,240,211,201,44,240,213
3340 DATA76,8,175,162,14,76,55
3350 DATA164,0,0,0,0,0,0
3360 DATA0,32,225,138,32,134,138
3370 DATA133,89,162,0,32,188,137
3380 DATA162,0,32,223,134,134,252
3390 DATA32,218,137,120,173,0,3
3400 DATA141,196,138,173,1,3,141
3410 DATA197,138,169,92,141,0,3
3420 DATA169,138,141,1,3,88,32
3430 DATA232,137,76,179,134,32,93
3440 DATA138,32,238,137,76,167,134
3450 DATA165,252,56,229,34,240,3
3460 DATA76,5,135,164,35,162,64
3470 DATA165,1,41,254,133,1,189
3480 DATA64,191,240,7,145,87,232
3490 DATA200,76,199,134,165,1,9
3500 DATA1,133,1,136,132,35,76
3510 DATA170,134,32,128,138,197,89
3520 DATA240,3,76,8,175,32,128
3530 DATA138,240,17,197,89,240,13
```

BASIC LOADER

```
3540 DATA157,128,191,232,224,64,208
3550 DATA239,162,23,76,55,164,169
3560 DATA0,157,128,191,96,160,0
3570 DATA177,87,133,20,200,177,87
3580 DATA133,21,162,0,200,196,35
3590 DATA240,10,177,87,157,0,2
3600 DATA232,224,86,208,241,165,1
3610 DATA41,254,133,1,160,0,185
3620 DATA128,191,240,9,157,0,2
3630 DATA232,200,224,87,208,242,165
3640 DATA1,9,1,133,1,165,35
3650 DATA24,101,34,168,165,35,24
3660 DATA101,252,133,35,198,35,177
3670 DATA87,157,0,2,200,232,201
3680 DATA0,240,10,224,88,208,241
3690 DATA169,0,157,0,2,232,142
3700 DATA185,135,138,24,105,4,133
3710 DATA11,173,2,3,141,186,135
3720 DATA173,3,3,141,187,135,169
3730 DATA136,141,2,3,169,135,141
3740 DATA3,3,32,139,138,164,11
3750 DATA76,164,164,173,186,135,141
3760 DATA2,3,173,187,135,141,3
3770 DATA3,32,165,138,165,87,197
3780 DATA45,208,6,165,88,197,46
3790 DATA240,19,173,185,135,201,1
3800 DATA240,3,76,170,134,160,2
3810 DATA132,35,162,0,76,173,134
3820 DATA76,75,138,0,0,0,165
3830 DATA46,133,96,165,45,133,95
3840 DATA56,229,47,165,96,229,48
3850 DATA144,3,76,167,136,32,127
3860 DATA136,165,37,240,43,201,1
3870 DATA240,71,201,2,240,92,169
3880 DATA37,32,210,255,32,157,136
3890 DATA169,61,32,210,255,160,2
3900 DATA177,95,72,200,177,95,168
3910 DATA104,32,145,179,32,221,189
3920 DATA32,30,171,76,99,136,169
3930 DATA32,32,210,255,32,157,136
3940 DATA169,61,32,210,255,32,133
3950 DATA177,165,71,164,72,32,162
3960 DATA187,32,221,189,32,218,189
3970 DATA76,99,136,32,157,136,169
3980 DATA47,160,136,32,30,171,76
3990 DATA99,136,32,61,32,70,85
4000 DATA78,67,84,73,79,78,0
4010 DATA162,3,189,132,137,32,210
4020 DATA255,224,3,208,3,32,157
4030 DATA136,202,16,240,160,4,177
4040 DATA95,133,35,136,177,95,133
4050 DATA34,136,177,95,32,36,171
4060 DATA169,34,32,210,255,169,13
4070 DATA32,210,255,32,225,255,208
4080 DATA1,96,24,165,95,105,7
4090 DATA133,95,166,96,144,1,232
4100 DATA134,96,76,196,135,160,0
4110 DATA132,37,200,177,95,10,38
4120 DATA37,74,153,69,0,136,16
4130 DATA244,165,69,32,210,255,165
4140 DATA70,240,3,32,210,255,96
4150 DATA165,70,208,5,169,32,32
4160 DATA210,255,96,169,13,32,210
4170 DATA255,165,47,133,95,165,48
4180 DATA133,96,165,96,197,50,208
4190 DATA6,165,95,197,49,240,173
4200 DATA32,225,255,240,168,32,127
4210 DATA136,165,37,240,10,201,2
4220 DATA208,3,169,36,44,169,37
4230 DATA44,169,32,32,210,255,32
4240 DATA157,136,169,32,32,210,255
4250 DATA169,40,32,210,255,165,95
4260 DATA24,105,3,133,251,165,96
4270 DATA105,0,133,252,160,1,177
4280 DATA251,133,253,169,0,133,254
4290 DATA6,253,38,254,165,253,24
4300 DATA101,251,133,253,165,254,101
4310 DATA252,133,254,160,0,177,253
4320 DATA141,131,137,200,177,253,141
4330 DATA130,137,208,3,206,131,137
4340 DATA206,130,137,173,131,137,174
4350 DATA130,137,164,95,140,130,137
4360 DATA164,96,140,131,137,32,205
4370 DATA189,172,130,137,132,95,172
4380 DATA131,137,132,96,56,165,253
4390 DATA233,2,133,253,165,254,233
4400 DATA0,133,254,197,252,208,6
4410 DATA165,253,197,251,240,8,169
4420 DATA44,32,210,255,76,16,137
4430 DATA160,3,177,95,133,251,136
4440 DATA177,95,24,101,95,133,95
4450 DATA165,96,101,251,133,96,169
4460 DATA41,32,210,255,169,13,32
4470 DATA210,255,76,180,136,0,0
4480 DATA34,32,61,36,32,134,138
4490 DATA133,89,162,0,32,188,137
4500 DATA32,218,137,120,173,0,3
4510 DATA141,196,138,173,1,3,141
4520 DATA197,138,169,92,141,0,3
4530 DATA169,138,141,1,3,88,32
4540 DATA232,137,32,93,138,32,238
4550 DATA137,76,176,137,76,8,175
4560 DATA32,128,138,240,248,197,89
4570 DATA240,13,157,64,191,232,224
4580 DATA64,208,239,162,23,76,55
4590 DATA164,169,0,157,64,191,134
4600 DATA34,96,165,43,24,105,2
4610 DATA133,87,165,44,105,0,133
4620 DATA88,96,162,0,160,2,132
4630 DATA35,165,1,41,254,133,1
4640 DATA177,87,240,33,221,64,191
4650 DATA8,165,1,9,1,133,1
```

BASIC LOADER

```
4660 DATA40,208,7,200,232,228,34
4670 DATA208,227,96,230,35,164,35
4680 DATA162,0,177,87,240,3,76
4690 DATA238,137,165,1,9,1,133
4700 DATA1,165,87,56,233,2,133
4710 DATA87,165,88,233,0,133,88
4720 DATA160,0,177,87,133,89,200
4730 DATA177,87,133,88,5,89,240
4740 DATA16,165,89,24,105,2,133
4750 DATA87,165,88,105,0,133,88
4760 DATA76,232,137,120,173,196,138
4770 DATA141,0,3,173,197,138,141
4780 DATA1,3,88,76,116,164,96
4790 DATA160,0,32,139,138,169,145
4800 DATA32,210,255,177,87,133,20
4810 DATA200,177,87,133,21,32,19
4820 DATA166,32,201,166,32,165,138
4830 DATA230,35,164,35,162,0,96
4840 DATA230,122,208,2,230,123,160
4850 DATA0,177,122,96,165,34,141
4860 DATA191,138,165,35,141,192,138
4870 DATA165,87,141,193,138,165,88
4880 DATA141,194,138,165,252,141,195
4890 DATA138,96,173,191,138,133,34
4900 DATA173,192,138,133,35,173,193
4910 DATA138,133,87,173,194,138,133
4920 DATA88,173,195,138,133,252,96
4930 DATA0,0,0,0,0,0,0
4940 DATA169,0,133,10,32,212,225
4950 DATA169,0,133,185,165,45,56
4960 DATA233,2,170,165,46,233,0
4970 DATA168,165,10,32,213,255,32
4980 DATA51,165,165,45,164,46,56
4990 DATA233,2,133,87,152,233,0
5000 DATA133,88,160,0,177,87,208
5010 DATA27,200,177,87,208,22,165
5020 DATA87,24,105,2,133,45,133
5030 DATA47,133,49,165,88,105,0
5040 DATA133,46,133,48,133,50,96
5050 DATA160,0,177,87,133,89,200
5060 DATA177,87,133,88,165,89,133
5070 DATA87,76,242,138,76,113,168
5080 DATA76,8,175,76,8,175,76
5090 DATA8,175,76,8,175,76,8
5100 DATA175,76,8,175,76,8,175
5110 DATA76,8,175,76,8,175,76
5120 DATA8,175,76,8,175,76,8
5130 DATA175,76,8,175,76,8,175
5140 DATA76,8,175,76,8,175,76
5150 DATA8,175,76,8,175,76,8
5160 DATA175,76,8,175,76,8,175
5170 DATA76,8,175,76,8,175,76,999
```


COLLINS – THE BEST IN COMPUTING

Advanced Commodore 64 BASIC Revealed
Nick Hampshire

Although it is relatively easy to learn to program the Commodore 64 in BASIC, advanced programmers need to know much more than how to use the fundamental commands. This book explains all these details and sets out a unique library of routines to add extra commands to BASIC. 0 00 383088 8 £9.95



Advanced Commodore 64 Graphics and Sound
Nick Hampshire

Any programmer wishing to make full use of the graphics and sound capability of the Commodore 64 will find this book an invaluable mine of essential information – much of it previously unpublished. 0 00 383089 6 £10.95

ORDER FORM
Please order from your bookseller or send this form with your remittance or credit card details to: Dept 44, Collins Professional and Technical Books, 8 Grafton Street, London W1X 3LA

Please send :
☐ ADVANCED COMMODORE 64 BASIC REVEALED 0 00 383088 8
☐ ADVANCED COMMODORE 64 GRAPHICS AND SOUND 0 00 383089 6

I enclose my remittance for £

(Postage: Add £1.50 for first purchase plus 50p for each additional item)

Name

Address

**** TAPE BACK-UP DEVICES ****

DCL – Interface	The DCL1 links two Datasettes, so that loading any program (including Turbos) from Datasette No. 1 simultaneously creates a BACK-UP TAPE on Datasette 2. No software needed. * Thousands sold * ORDER AS DCL 1 PRICE £10.00
DCL4 Interface	This SUPER DCL1 is for those with one dtasette and one audio recorder. Back-up tapes can be made on either a second Datasette or on an audio recorder. A LED data monitor shows the start and end of programs. * Very Popular * ORDER AS DCL4 PRICE £17.00
DCL4A Interface	SIMILAR to the DCL4 but fitted with an Audible data monitor. ** NEW ** ORDER AS DCL4A PRICE £18.50

**** BACK-UP PROBLEMS SOLVED ****

Alignment Tape	Realign your Datasette tape head VERY PRECISELY with this easy to use kit. (No dismantling of the Datasette required.) Two high baud rate tests are provided plus a special screwdriver and full instructions. ** NEW ** ORDER AS DHA1 PRICE £7.50
Tape Head Demagnetiser	This mains powered demagnetiser, if used a few seconds per week (no need to dismantle the Datasette), will reduce troublesome residual magnetism and improve loadability. * Indispensable * ORDER AS THD1 PRICE £6.30

**** GT LOADER CARTRIDGE ** for 1541 D/Drive ****

GT Loader Cartridge	This cartridge enables your 1541 d/drive to load 4-5 times faster. With many useful features including abbreviated load/save commands and unique on/off switching, etc. * Essential * ORDER AS GTL1 PRICE £20.00 ALSO AVAILABLE WITH BUILT-IN RESET SWITCH. ORDER AS GTL2 PRICE £22.00

**** FREE CATALOGUE ** Please send 17p stamp**

Free Catalogue	Full details of these and many other products all at... ** FULLY INCLUSIVE PRICES ** eg BROTHER HR5 80 col Dot Matrix Printer.........£149.00 eg FIDELITY CM14 Colour Monitor for CBM64......£199.00 eg PRINTER Cable and software driver for CBM64..£19.90 eg THE STICK. The fantastic new baseless joystick.£13.95 PLUS Interfaces, cleaning kits, printer/monitor leads, etc. etc.

**** 10 DAY MONEY BACK GUARANTEE ON ALL HARDWARE ****

Terms:	ORDERING: ALL PRICES ARE FULLY INCLUSIVE – NO HIDDEN EXTRAS AND INCLUDE RECORDED DELIVERY. (EXPORT: add £1.00 extra)

TRILOGIC
Dept YC
(formerly Trigsoft)
29 HOLEM LANE, BRADFORD
BD4 0QA
TEL: 0274 684289

Your Commodore's man in the States, Burton Rubin, attended the launch of the Amiga. Here's his report plus Brendin Lewis's impressions of Metacomco — makers of Amigados.

THE AMIGA U

US View

FOR THE PAST TWO YEARS, WISPS OF rumours, as insubstantial as smoke, have appeared in various magazines concerning the fabled, long awaited, Amiga computer. Originally called the Lorraine, the product of the Amiga Corporation, the machine, a 32 bit computer with fabulous graphic and sound capability, was to be the ultimate home computer.

Silicon Valley, as we all know, is littered with the bones of "ultimate home computers". Timex-Sinclair, Texas Instruments, Coleco, Mattel, and even the mighty IBM have all seen their offereings for the home market wither away to a dusty death. When Jack Tramiel left Commodore, there was some speculation that his plan might be to purchase Amiga, (and with it, the rights to the wonderous Lorraine) and use the new machine as the saviour of the almost moribund Atari. By purchasing, Amiga, Commodore beat Atari to the punch.

Commodore first exhibited the machine at the June, 1984 Consumer



The Amiga Specifications

Machine:	Commodore Amiga
Processor:	68000 at 7MHz
RAM:	256K standardd; 256K expansion slot; Up to Mb
ROM:	192K operating system.
Graphics:	320 × 200 32 colours; 320 × 400 32 colours 640 × 200 16 colours; 640 × 400 16 colours Colours selected from a palette of 4096
Sound:	Four audio channels including stereo and speech
Disk:	3½ inch floppy 880Kb
Software:	Amigados, speech, Basic, Graphics
Miscellaneous:	Peripherals: Videodisk unit, hard disk plus 1M byte RAM, colour monitor, modem. Three custom chips inside the Amiga take much of the processing burden away from the 68000. The very high speed graphics are handled by one of these chips.

UNVEILED

Electronics show and its specifications were impressive. The Motorola 68000 was chosen for the central processing unit. This is a 32 bit chip with a 16 bit bus. Capable of addressing up to 16 megabytes of memory, it is the same chip that powers the Apple Macintosh. Tramiel's riposte was the 130ST and 520ST, 32 bit machines running GEM. The Commodore camp responded with. . .silence. Sure, there was plenty on the sparkling, rational new 128, and even pictures and publicity on the new notebook computer. But Commodore breathed nary a word about the Lorraine, and the rumours continued to fly.

At the official premier of the Amiga computer, hosted (quite grandly) by Commodore at the Vivian Beaumont theatre at Lincoln Centre, the rumours all came true.

The specifications do not convey the power of the machine. The standard configuration of the Amiga includes a Motorola 68000 microprocessor, running at almost eight MHz, with 256K of RAM, internally expandable to 512K. The speed of this microprocessor is further abetted by a proprietary three chip set which frees the 68000 from routine graphic and I/O tasks.

The Amiga is controlled by an 89 key keyboard with numeric keypad, cursor and special function keys, or a two button mouse. It features a built in three and a half inch disk drive (880K formatted), 80 × 25 line text display, 640 × 400 resolution and a palette of 4,096 colours (of which, any 16 can be on the screen at one time in high resolution mode). There are parallel, serial, and second drive ports, two reconfigurable joystick ports, as well as text to voice and professional quality four channel multi-voice music synthesis capability. ABasiC, Amiga ODS, and Amiga Tutor will be bundled with the machine.

Software availability, the bane of all new computer introductions, would seem not to be a problem here. Thanks to the Emulator – a software option – the Amiga can run IBM PC compatible software packages like Lotus 1-2-3. Wordstar, and D Base III, in either 3.5 or 5.25 inch disk format.

In addition, when introduced in September, the Amiga will have available more than 20 programs including word processing, accounting, productivity, education, speech synthesis, telecommunications, paint, animation, and graphic programs. Assembler, C, Pascal, Logo, and Lisp will be available at the time of introduction, as well as a very powerful version of Microsoft Basic. Third party publishers will have more than 20 games available when the machine hits the shelves. Arktronics has already completed work on a videodisk interface.

The Amiga works through the now familiar system of menus, windows, icons, multitasking, and a mouse – first pioneered by Apple with the Lisa. Up to 50 windows can be open and running on the Amiga, though this is obviously more than anyone can manipulate without going insane. The Intuition operating system, working through a system of "gadgets" makes windowing an easier and quicker task than it is on the Macintosh, or with the GEM operating system of the Atari.

The animation power of the machine is almost beyond description. It should open new vistas for anyone who needs professional visual aids. Its value to the small advertising agency, or any small business should be incalculable. Combined with the phenomenal multitasking capability, Amiga should be a formidable weapon in the business wars.

With business applications firmly in mind, Commodore has chosen to merchandise the machine through specialist dealers. Long ago, Commodore pulled the plug on computer stores, in favour of mass market merchants.

The powers at Commodore are reversing the decision that made them such a commercial success and critical failure. It remains to be seen just how easily Commodore will be accepted by the same computer stores which were abandoned a few years ago.

At a list price of $1295, the Amiga represents excellent value. However, it's unlikely to be found at the local K Mart.

My feeling is that acceptance in the computer stores may come gradually, but it will definitely come. The machine is simply too good to be ignored. The $1295 price tag includes a healthy mark-up for the dealers, and Commodore has signed up 800 RCA service locations to provide service support. With the advent of the Amiga, Commodore has both Apple and IBM lined up in its sights.

Thomas Rattigan, president of Commodore North America, is talking tough. "Commodore" he says, "is a strong, lean, aggressive, organization", and he intends to have the Amiga showcased in 10,000 outlets within a year of its introduction. He feels that the Amiga is a great leap forward (shades of Sir Clives QL!).

Irving Gould, Chairman of the company, sees the future of Commodore in "sophisticated, high end systems", with "excellent price/performance ratios" and "a full, rich, product line". That doesn't sound like the marketing philosophy that we've all grown to know and love.

The Amiga, though, is a machine capable of changing the philosophy of a company. Where the trusty old 64 was a Ford, and the Plus Four an Edsel, the Amiga is a Ferrari. The ultimate decisions will be made in the marketplace. Don't bet against the Amiga.

The UK Connection

Tucked away in the corner of a small square in Bristol is the software house Metacomco. What is Metacomco? It is the company which has written Amigados, the operating system for the new Commodore Amiga. Metacomco is not a large company – with a staff of 25 – but it does have a good track record, working on software for both the Sinclair QL and the Atari 520ST.

Upon meeting a few of the staff it's quite easy to see that the firm's success is based on three main factors – sound management, expertise and, most notably, enthusiasm. Even some members of the senior management seem like small children playing with a new toy whenever the Amiga is mentioned. Even this cynical reporter was surprised at its performance. Though, as ever I'm still not totally convinced, I'll reserve final judgement for the full production model. Metacomco's staff, on the other hand, have had the word cynical totally erased from their memories. It was difficult for them to see a market into which the Amiga would not fit.

The whole story really started about three years ago when Amiga Inc. started work on a new machine. In November 1984, Commodore took over Amiga and thus the machine. Previous to this though, Commodore had already approached Metacomco concerning Tripos (which is the framework around which Amigados is built). From here, Metacomco has never looked back and has written various bits of software for the Amiga, including a version of Basic.

Although, for most of the day it was difficult to stop our host talking, I did finally get an ominous silence when I broached the subject of Amiga II. The only reply I did get was that, due to the open technology of the Amiga, the Japanese would be the first to produce an Amiga look-a-like and that ideas were already in the pipeline for something within the next 12 months.

Finally, my thanks to all at Metacomco for a very enjoyable day, and for providing what Commodore UK could not (or would not), a look at the Amiga.

SPLIT 64

Split your 64's memory with this useful little program from Ray Green.

Introduction

WHEN USING YOUR C64 TO develop Basic programs there are times when it would be useful to have another machine set up close to hand in order to run small utility programs, for instance hex conversion, address calculation, etc. In schools, two pupils sharing one machine must be working on the same program unless a very disciplined approach is used. Split 64 is my attempt to solve these problems without incurring the cost of an extra machine.

Method

The program splits Basic memory into two areas of just over 16K allowing the machine to hold two different Basic programs at the same time. Switching between areas is achieved by holding down the shift key and pressing the control key. Each area maintains the screen information from the last time you used it. The variable contents also remain intact. The second screen information is stored at the top of memory leaving approximately 3K free for things like centronics interface software. The area from $C000 is also free for utilities.

In the initial start up of the program, I set the screen colours up to the normal blue for area (a) and green for area (b). If these are not to your liking then just change them with the normal POKEs and the colours you set will be stored when switching between areas. If while using this program, run/stop restore is pressed, then as normal the screen will reset to blue and clear. However, the shift control switch will no longer work so to re-start Split 64 without losing your programs, type SYS (35896) [return]. SYS (35840) resets split 64 completely.

Basic Loader

```
1 REM ***********************************
2 REM *                                 *
3 REM *  R. GREEN     1/2/85            *
4 REM *                                 *
5 REM ***********************************
6 A=35840
10 PRINT"[CLEAR][DOWN][RIGHT][RIGHT][RIG
HT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][R
IGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT][RVSON]SPLIT 64[RVSOFF]"
15 PRINT"[DOWN][DOWN][DOWN][DOWN][DOWN][
DOWN][DOWN][DOWN][DOWN][DOWN][RIGHT][RIG
HT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][R
IGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT] WORKING"
20 READD
25 C=C+D
30 IFD<0THEN40
35 POKEA,D:A=A+1:GOTO20
40 IFC=37269THENSYS(35840)
45 PRINT"[DOWN][DOWN]ERROR IN DATA"
50 DATA162,16,189,46,141,141,134,2,202,1
89
55 DATA46,141,141,33,208,202,189,46,141,
141
60 DATA32,208,202,189,46,141,149,43,202,
16
65 DATA248,162,2,169,0,157,0,8,157,0
70 DATA74,202,16,247,32,34,228,32,120,14
0
75 DATA32,34,228,32,120,140,120,173,20,3
80 DATA72,173,21,3,72,169,87,141,20,3
85 DATA169,140,141,21,3,104,141,81,141,1
04
90 DATA141,80,141,88,76,116,164,173,141,
2
95 DATA201,5,208,23,165,207,208,19,165,1
57
100 DATA201,128,208,13,32,120,140,162,25
5,160
105 DATA255,136,208,253,202,208,248,108,
80,141
```

BASIC Loader (continued)

```
110 DATA162,16,173,134,2,72,189,63,141,1
41
115 DATA134,2,104,157,63,141,202,173,33,
208
120 DATA72,189,63,141,141,33,208,104,157
,63
125 DATA141,202,173,32,208,72,189,63,141
,141
130 DATA32,208,104,157,63,141,202,181,43
,72
135 DATA189,63,141,149,43,104,157,63,141
,202
140 DATA16,241,162,0,189,0,4,72,189,82
145 DATA141,157,0,4,104,157,82,141,189,0
150 DATA216,72,189,93,145,157,0,216,104,
157
155 DATA93,145,189,0,5,72,189,82,142,157
160 DATA0,5,104,157,82,142,189,0,217,72
165 DATA189,93,146,157,0,217,104,157,93,
146
170 DATA189,0,6,72,189,82,143,157,0,6
175 DATA104,157,82,143,189,0,218,72,189,
93
180 DATA147,157,0,218,104,157,93,147,189
,0
185 DATA7,72,189,82,144,157,0,7,104,157
190 DATA82,144,189,0,219,72,189,93,148,1
57
195 DATA0,219,104,157,93,148,232,224,0,2
08
200 DATA139,96,1,8,3,8,3,8,3,8
205 DATA0,74,0,74,0,74,254,246,14,1
210 DATA74,3,74,3,74,3,74,0,140,0
215 DATA140,0,140,245,240,13,0,0,-1
```

Entering the Program

To enter the program type in the Basic loader program, then SAVE it. Type RUN and the screen should flash, then clear to the normal Basic start up message. Note the much reduced free memory. Type in a short Basic program and RUN it. Stop the program, hold down the shift key and press control. The screen should now change colour and have the Basic start up message on it.

You are now in the second area. LIST and there should be no trace of the program you typed in. Shift control should take you back to the first screen and LIST will show your program is still there. If you wish to make a machine code copy of split 64 then first make sure you are in program area (a) before entering your machine code monitor. The start address is $8C00 and the end address $8D52.

How it Works — General

Split 64 takes advantage of the fact that the C64 operating system allows Basic to work within any free area of memory. To write your programs in another area of memory it is only necessary to change the pointers used by Basic in zero page. Change the pointers back and any program that was there before will still be there. It is also possible to change the area of memory used by the screen but when writing this program it was decided to store the second screen along with the second colour map above Basic memory. The shift control switch is operated by a simple interrupt wedge.

How it Works — Machine Code

Initialise Routine

This routine sets up area (a) colours, memory start, memory end, and variable pointers from values fixed in Table 1. It then puts zero in the first three locations of Basic memory for areas (a) and (b). It then calls the swap routine, which stores the current screen and loads the contents of the second screen (rubbish at this time). Having swapped screens the Basic start up program is called which clears the screen and prints the start up message. This procedure is then repeated for area (a). Finally the wedge routine address is inserted into the IRQ vector, the normal vector address having been saved, then back to basic.

Wedge Routine

The wedge routine is entered on every IRQ. It first checks location $028D to see if the shift control keys are pressed. If not it jumps to the normal IRQ routine. If the keys are pressed, it checks that the cursor is off and that the computer is not in RUN mode. When all three of these conditions are satisfied it calls the swap routine. After swapping areas a large delay loop is executed to avoid multiple swaps. Control is then passed to the normal IRQ routine.

Swap Basic

This is the first part of the swap subroutine. A loop is used to exchange the current screen, border and character colours, with the contents of Table 2. The same loop is used to exchange the Basic pointers ($2B-$38) again with the contents of Table 2. On start up Table 2 contains the fixed values for area (b).

Swap Screen

Once more a loop is employed to exchange the current screen and colour map stored above Basic memory. The exchange is done eight bytes at a time. Not the most elegant way to swap four areas of memory but it was chosen to cause minimum disruption to the screen during the swap.

Table 1

```
INITIALISE ROUTINE

8c00 a2 10       ldx #$10       :index to table1
8c02 bd 2e 8d    lda $8d2e,x    :table1
8c05 8d 86 02    sta $0286      :char. colour area a
8c08 ca          dex            :next
8c09 bd 2e 8d    lda $8d2e,x    :table1
8c0c 8d 21 d0    sta $d021      :screen colour area a
8c0f ca          dex            :next
8c10 bd 2e 8d    lda $8d2e,x    :table1
8c13 8d 20 d0    sta $d020      :border colour area a
8c16 ca          dex            :next
8c17 bd 2e 8d    lda $8d2e,x    :table1
8c1a 95 2b       sta $2b,x      :basic memory size area a
8c1c ca          dex            :
8c1d 10 f8       bpl $8c17      :next
8c1f a2 02       ldx #$02       :
8c21 a9 00       lda #$00       :zero first 3 bytes in
8c23 9d 00 08    sta $0800,x    :both area a&b
8c26 9d 00 4a    sta $4a00,x    :
8c29 ca          dex            :
8c2a 10 f7       bpl $8c23      :
8c2c 20 22 e4    jsr $e422      :basic start up a
8c2f 20 78 8c    jsr $8c78      :swap area
8c32 20 22 e4    jsr $e422      :basic start up b
8c35 20 78 8c    jsr $8c78      :swap back
8c38 78          sei            :
8c39 ad 14 03    lda $0314      :irq low
8c3c 48          pha            :save it
8c3d ad 15 03    lda $0315      :irq high
8c40 48          pha            :save it
8c41 a9 57       lda #$57       :low address wedge
8c43 8d 14 03    sta $0314      :irq vector
8c46 a9 8c       lda #$8c       :high address wedge
8c48 8d 15 03    sta $0315      :irq vector
8c4b 68          pla            :recover address irq high
8c4c 8d 51 8d    sta $8d51      :store
8c4f 68          pla            :recover address irq low
8c50 8d 50 8d    sta $8d50      :store
8c53 58          cli            :
8c54 4c 74 a4    jmp $a474      :ready for basic
```

Table 2

```
WEDGE ROUTINE


8c57 ad 8d 02     lda $028d       :control shift pressed
8c5a c9 05        cmp #$05        :
8c5c d0 17        bne $8c75       :if not goto normal irq
8c5e a5 cf        lda $cf         :cursor off
8c60 d0 13        bne $8c75       :if not goto normal irq
8c62 a5 9d        lda $9d         :direct mode
8c64 c9 80        cmp #$80        :
8c66 d0 0d        bne $8c75       :if not goto normal irq
8c68 20 78 8c     jsr $8c78       :swap area
8c6b a2 ff        ldx #$ff        :
8c6d a0 ff        ldy #$ff        :
8c6f 88           dey             :delay to avoid key bounce
8c70 d0 fd        bne $8c6f       :
8c72 ca           dex             :
8c73 d0 f8        bne $8c6d       :
8c75 6c 50 8d     jmp ($8d50)     :jump to normal irq routine
```

Table 3

```
SWAP BASIC Z.PAGE LOCATIONS

8c78 a2 10        ldx #$10        :index to table2
8c7a ad 86 02     lda $0286       :char colour current
8c7d 48           pha             :save it
8c7e bd 3f 8d     lda $8d3f,x     :table2
8c81 8d 86 02     sta $0286       :change char colour
8c84 68           pla             :recover char colour
8c85 9d 3f 8d     sta $8d3f,x     :store in table2
8c88 ca           dex             :next
8c89 ad 21 d0     lda $d021       :current screen colour
8c8c 48           pha             :save it
8c8d bd 3f 8d     lda $8d3f,x     :table2
8c90 8d 21 d0     sta $d021       :change screen colour
8c93 68           pla             :recover screen colour
8c94 9d 3f 8d     sta $8d3f,x     :store in table2
8c97 ca           dex             :next
8c98 ad 20 d0     lda $d020       :current border colour
8c9b 48           pha             :save it
8c9c bd 3f 8d     lda $8d3f,x     :table2
8c9f 8d 20 d0     sta $d020       :change border colour
```

Table 3 continued

```
8ca2 68         pla           :recover border colour
8ca3 9d 3f 8d   sta $8d3f,x   :store in table2
8ca6 ca         dex           :next
8ca7 b5 2b      lda $2b,x     :basic memory size
8ca9 48         pha           :save
8caa bd 3f 8d   lda $8d3f,x   :table2
8cad 95 2b      sta $2b,x     :basic memory size
8caf 68         pla           :recover
8cb0 9d 3f 8d   sta $8d3f,x   :store in table2
8cb3 ca         dex           :next
8cb4 10 f1      bpl $8ca7     :repeat for other z.page
                               locations
```

Table 4

```
SCREEN SWAP ROUTINE

8cb6 a2 00      ldx #$00      :index for screen+colour mem
8cb8 bd 00 04   lda $0400,x   :screen mem
8cbb 48         pha           :save it
8cbc bd 52 8d   lda $8d52,x   :second screen
8cbf 9d 00 04   sta $0400,x   :store in screen
8cc2 68         pla           :recover
8cc3 9d 52 8d   sta $8d52,x   :store in second screen
8cc6 bd 00 d8   lda $d800,x   :colour mem
8cc9 48         pha           :save it
8cca bd 5d 91   lda $915d,x   :second colour mem
8ccd 9d 00 d8   sta $d800,x   :store in colour mem
8cd0 68         pla           :recover
8cd1 9d 5d 91   sta $915d,x   :store in second colour mem
8cd4 bd 00 05   lda $0500,x   : --------------------------
8cd7 48         pha           :
8cd8 bd 52 8e   lda $8e52,x   :
8cdb 9d 00 05   sta $0500,x   :
8cde 68         pla           :
8cdf 9d 52 8e   sta $8e52,x   :     same but plus 256
8ce2 bd 00 d9   lda $d900,x   :
8ce5 48         pha           :
8ce6 bd 5d 92   lda $925d,x   :
8ce9 9d 00 d9   sta $d900,x   :
8cec 68         pla           :
8ced 9d 5d 92   sta $925d,x   : --------------------------
8cf0 bd 00 06   lda $0600,x   :
8cf3 48         pha           :
```

Table 4 (continued)

```
8cf4 bd 52 8f     lda $8f52,x    :
8cf7 9d 00 06     sta $0600,x    :
8cfa 68           pla            :    same but plus 512
8cfb 9d 52 8f     sta $8f52,x    :
8cfe bd 00 da     lda $da00,x    :
8d01 48           pha            :
8d02 bd 5d 93     lda $935d,x    :
8d05 9d 00 da     sta $da00,x    :
8d08 68           pla            :
8d09 9d 5d 93     sta $935d,x    : -------------------------
8d0c bd 00 07     lda $0700,x    :
8d0f 48           pha            :
8d10 bd 52 90     lda $9052,x    :
8d13 9d 00 07     sta $0700,x    :
8d16 68           pla            :    same but plus 768
8d17 9d 52 90     sta $9052,x    :
8d1a bd 00 db     lda $db00,x    :
8d1d 48           pha            :
8d1e bd 5d 94     lda $945d,x    :
8d21 9d 00 db     sta $db00,x    :
8d24 68           pla            :
8d25 9d 5d 94     sta $945d,x    : -------------------------
8d28 e8           inx            :next
8d29 e0 00        cpx #$00       :255 done in each block
8d2b d0 8b        bne $8cb8      :if not then go back
8d2d 60           rts            :return
```

Table 5

```
TABLES

Contents shown as at start.


Table1                          Table2

8d2e 01 :low s.a                8d3f 01
8d2f 08 :high "                 8d40 4a
8d30 03 :low e.a                8d41 03
8d31 08 :high "                 8d42 4a
8d32 03 :low s.arrays           8d43 03
8d33 08 :high     "             8d44 4a
8d34 03 :low e.arrays           8d45 03
8d35 08 :high     "             8d46 4a

8d36 00 :low b/active string    8d47 00
8d37 4a :high   "          "    8d48 8c
8d38 00 :low t/ "          "    8d49 00
8d39 4a :high   "          "    8d4a 8c
8d3a 00 :low mem top            8d4b 00
8d3b 4a :high "    "            8d4c 8c
8d3c fe :screen colour          8d4d f5
8d3d f6 :border colour          8d4e f0
8d3e 0c :chai colour            8d4f 0d

Address for IRQ.

8d50 31
8d51 ea
```

This month Garry Marshall shows you how to create artificial landscapes on your 64.

PROGRAMMING PROJECTS

Graphic landscapes

The project

COMPUTER-GENERATED GRAPHICS are increasingly finding their way into films. Among the most impressive and realistic effects that have been seen are the entirely artificial landscapes created for some of the *Star Wars* films. This month's project involves the creation of an artificial landscape of this kind.

Of course, far more sophisticated (and expensive!) equipment than a C64 is needed to display graphics of a quality that is suitable for use in films. But, as we shall see, it is not too difficult to write programs that produce the basic effects, and the quality of the results is surprisingly good.

The theory the provides the basis for the creation of these landscapes is known as fractal geometry. Without going too deeply into the theory, the idea behind a fractal curve is that it is not one or two-dimensional, but has a dimension which is a fraction. Curves of this kind describe, for example, rough surfaces, with the degree of roughness determining the fraction giving the dimension of the fractal curve. They also describe many other naturally occurring curves, such as coast lines and hillsides, and this is what makes them eminently suitable for simulating landscapes.

The project involves starting with a triangle and, by means of a simple procedure deriving from fractal geometry, sub-dividing it to give a graphic display that resembles a mountain. The further the sub-division process is carried out, the more realistic the result will appear.

The solution

The basic idea of the procedure for creating the mountain landscape is illustrated in Figures 1 and 2. The first figure shows that taking a triangle, finding the mid-points of its sides and joining them gives four smaller triangles. Repeating the process on each of the smaller triangles gives a finer triangular mesh, and the more it is repeated, the

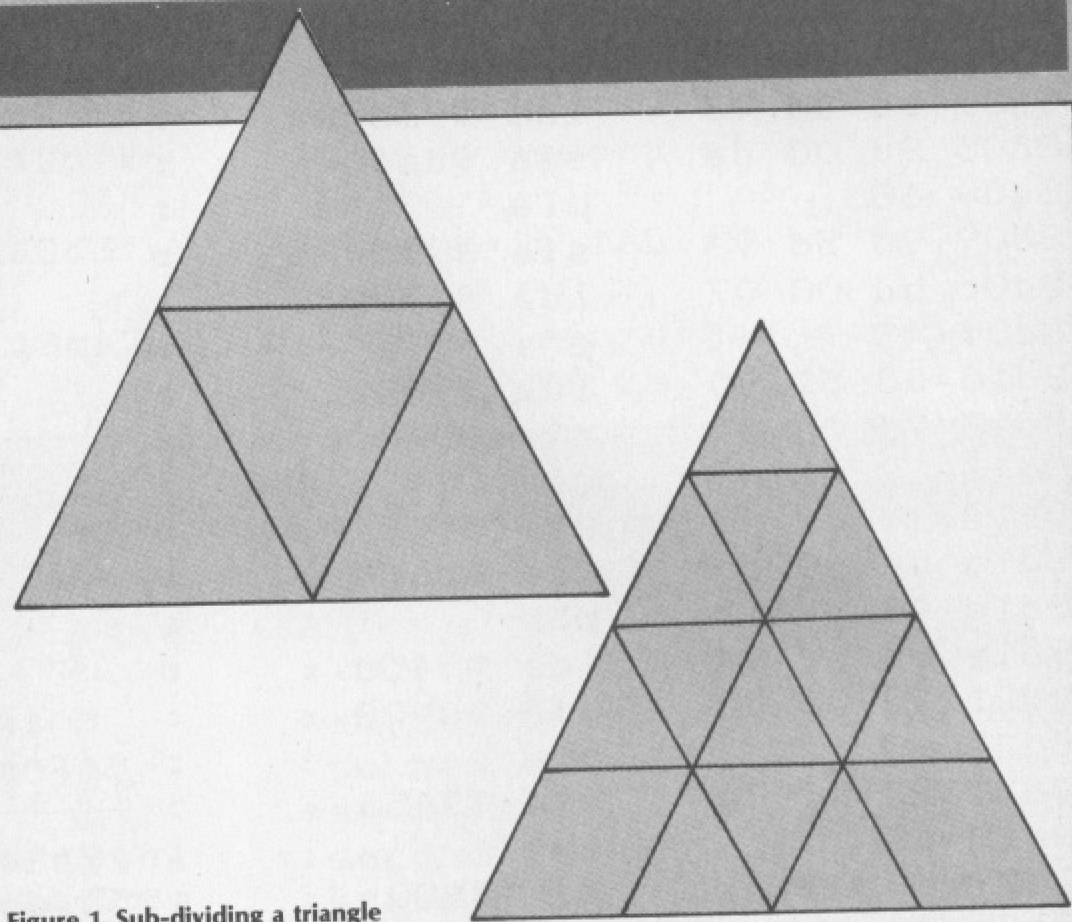


Figure 1. Sub-dividing a triangle

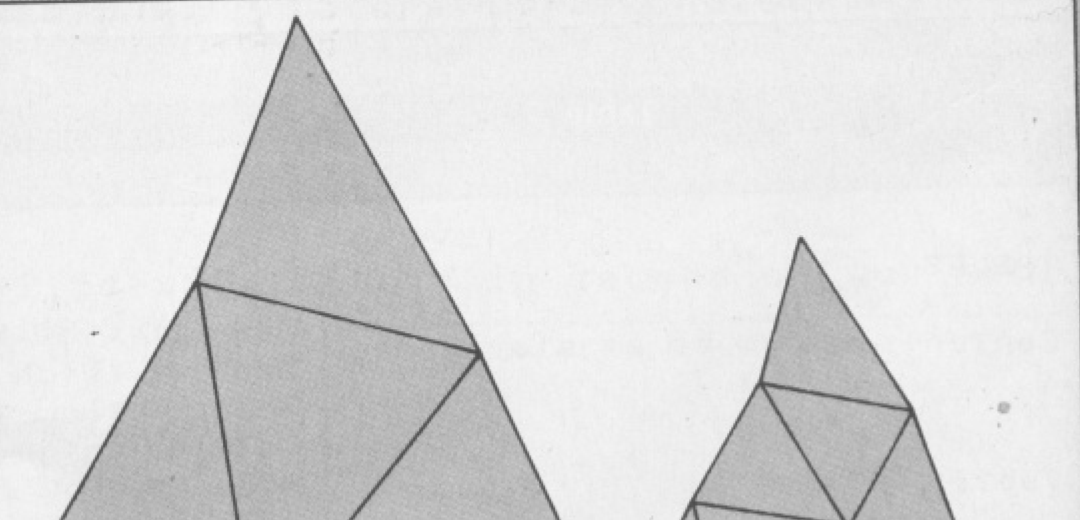


Figure 2. Sub-dividing and perturbing

finer the triangular mesh becomes. But the results of this procedure do not resemble any natural phenomenon, simply because of their regularity. This is not a property to be found in nature, which is characterised by randomness.

We can produce an element of randomness by perturbing the mid-point of each side by a random amount the variation of which is proportional to the length of its side. The left-hand illustration in Figure 2 shows the result of doing this and drawing the four smaller triangles produced by one sub-division. The right-hand illustration shows what happens after two sub-divisions, demonstrating that none of the triangles can be drawn until all the mid-points have been found and then perturbed. Continuing this process will give an increasingly mountain-like display. The range of the perturbation for the mid-point of each side or, if you like, the degree of proportionately between the range and the length of the side, effectively determines the fractional dimension of the final result. Different values for the range give quite different appearances to the mountain landscapes.

This gives us the following form for a graphics program to draw an artificial mountain landscape. It should declare the arrays to be used, and there will be several, because we must store the points for all the triangles in the final display as we cannot draw any of them until all their positions have been computed. Then it must read the co-ordinates of the corners of the basic triangle, prepare the high-resolution graphics screen, and carry out the sub-division of each side to give the points at the corners of the resulting triangles. To begin with, we will ignore the perturbations of the mid-points that introduce the randomness. This simplifies the program a little, gives results such as those in Figure 1, but provides a program which we can easily generalise to produce results like those in Figure 2. Finally, when the program has found the points for all the triangles, it only remains to plot them.

The main program based on this scheme is:

```
10 DIMX(48), Y(48), S(3), T(3), U(12), V(12),
XT(48), YT(48)
20 FOR K=1 TO 3
30 READ X(K), Y(K)
40 NEXT K
50 DATA 150, 50, 250, 150, 50, 150
60 GOSUB500: REM HI-RES SCREEN
70 FOR N=0 TO 1
80 FOR K=1 TO 4 N
90 GOSUB 4000: REM SUB-DIVIDE THE
TRIANGLES
100 NEXT K
110 FOR K=1 TO 4 (N+1)*3
120 X(K)=XT(K): Y(K)=YT(K)
130 NEXT K
140 NEXT N
```

Program Listing

```
10 DIMX(48), Y(48), S(3), T(3), U(12), V(12),
XT(48), YT(48)
20 FOR K=1 TO 3
30 READ X(K), Y(K)
40 NEXT K
50 DATA 150, 50, 250, 150, 50, 150
60 GOSUB500: REM HI-RES SCREEN
70 FOR N=0 TO 1
80 FOR K=1 TO 4^N
90 GOSUB 4000: REM  SUB-DIVIDE THE TRIANGLES
100 NEXT K
110 FOR K=1 TO 4^(N+1)*3
120 X(K)=XT(K): Y(K)=YT(K)
130 NEXT K
140 NEXT N
150 FOR K=1 TO 16
160 FOR J=1 TO 3
170 S(J)=X(J+3*(K-1)): T(J)=Y(J+3*(K-1))
180 NEXT J
190 GOSUB 3000: REM PLOT EACH TRIANGLE
200 NEXT K
210 END
500 POKE 53272, PEEK(53272) OR 8
510 POKE 53265, PEEK(53265) OR 32
520 FOR I=8192 TO 16192: POKE I, 0: NEXT I
530 FOR I=1024 TO 2023: POKE I, 22: NEXT I
540 RETURN
1000 RO=INT(R/8): CO=INT(C/8)
1010 L=R AND 7
1020 BIT=7 - (C AND 7)
1030 BYTE=8192 + RO*320 + CO*8 + L
1040 POKE BYTE, PEEK(BYTE) OR 2^BIT
1050 RETURN
2000 DX=X2-X1: DY=Y2-Y1
2010 IF DX=0 THEN 2070
2020 FOR C=X1 TO X2 STEP SGN(DX)
2030 R=INT(Y1+(C-X1)*DY/DX)
2040 GOSUB 1000: REM PLOT POINT
2050 NEXT C
2060 RETURN
2070 C=X1
2080 FOR R=Y1 TO Y2 STEP SGN(DY)
2090 GOSUB 1000: REM PLOT POINT
2100 NEXT R
2110 RETURN
3000 X1=S(3): Y1=T(3)
3010 FOR P=1 TO 3
3020 X2=S(P): Y2=T(P)
3030 GOSUB 2000: REM DRAW LINE
3040 X1=X2: Y1=Y2
3050 NEXT P
3060 RETURN
4000 FOR M=1 TO 3
4010 S(M)=X(M+3*(K-1)): T(M)=Y(M+3*(K-1))
4020 NEXT M
```

```
150 FOR K=1 TO 16
160 FOR J=1 TO 3
170 S(J)=x(J+3*(K-1)): T(J)=Y(J+3*Æ(K-1))
180 NEXT J
190 GOSUB 3000: REM PLOT EACH TRIANGLE
200 NEXT K
210 END
```

Here, the arrays are declared in line 10. The data, which is at line 50 and gives the positions of the corners of the initial triangle, is read by lines 20 to 40. Line 60 calls the now-familiar subroutine, starting at line 500, for preparing the high-resolution graphics screen.

Lines 70 to 140 calculate and store the positions of the corners of all the triangles that result from the sub-dividing process. The outer loop variable, N, determines how many stages of sub-division occur. In the program as presented, there are two stages, which are necessary to bring us to the position shown in the right-hand illustration of Figure 1. The first sub-division is done with N=0 and the second with N=1.

The inner loop variable, K, counts the number of triangles to be sub-divided. Initially, there will be one, and the value of 4^N when N=0 is 1. After the first subdivision, there will be four, the value

Program Listing (cont.)

```
4030 GOSUB 5000: REM SUB-DIVIDE THIS TRIANGLE
4040 FOR M=1 TO 12
4050 XT(M+12*(K-1))=U(M): YT(M+12*(K-1))=V(M)
4060 NEXT M
4070 RETURN
5000 A1=(S(1)-S(2))*0.2*(RND(0)-0.5)
5010 B1=(S(1)-S(3))*0.2*(RND(0)-0.5)
5020 C1=(S(2)-S(3))*0.2*(RND(0)-0.5)
5030 A=0.5*(S(1)+S(2))+A1
5040 D=0.5*(T(1)+T(2))+A1
5050 B=0.5*(S(1)+S(3))+B1
5060 E=0.5*(T(1)+T(3))+B1
5070 C=0.5*(S(2)+S(3))+C1
5080 F=0.5*(T(2)+T(3))+C1
5090 U(1)=S(1): U(4)=S(2): U(7)=S(3)
5100 V(1)=T(1): V(4)=T(2): V(7)=T(3)
5110 U(2)=A: U(6)=A: U(12)=A
5120 V(2)=D: V(6)=D: V(12)=D
5130 U(3)=B: U(8)=B: U(11)=B
5140 V(3)=E: V(8)=E: V(11)=E
5150 U(5)=C: U(9)=C: U(10)=C
5160 V(5)=F: V(9)=F: V(10)=F
5170 RETURN

READY.
```

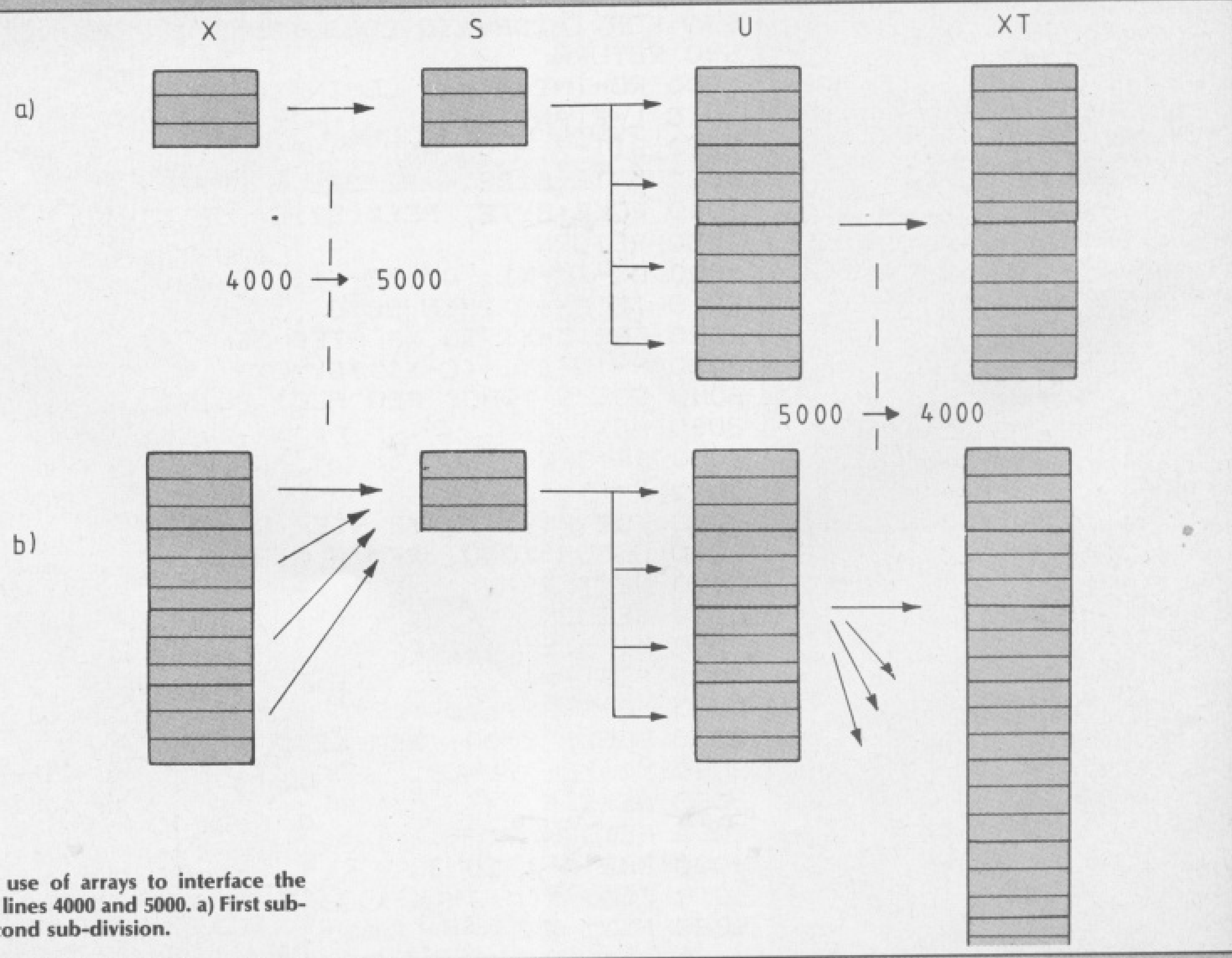
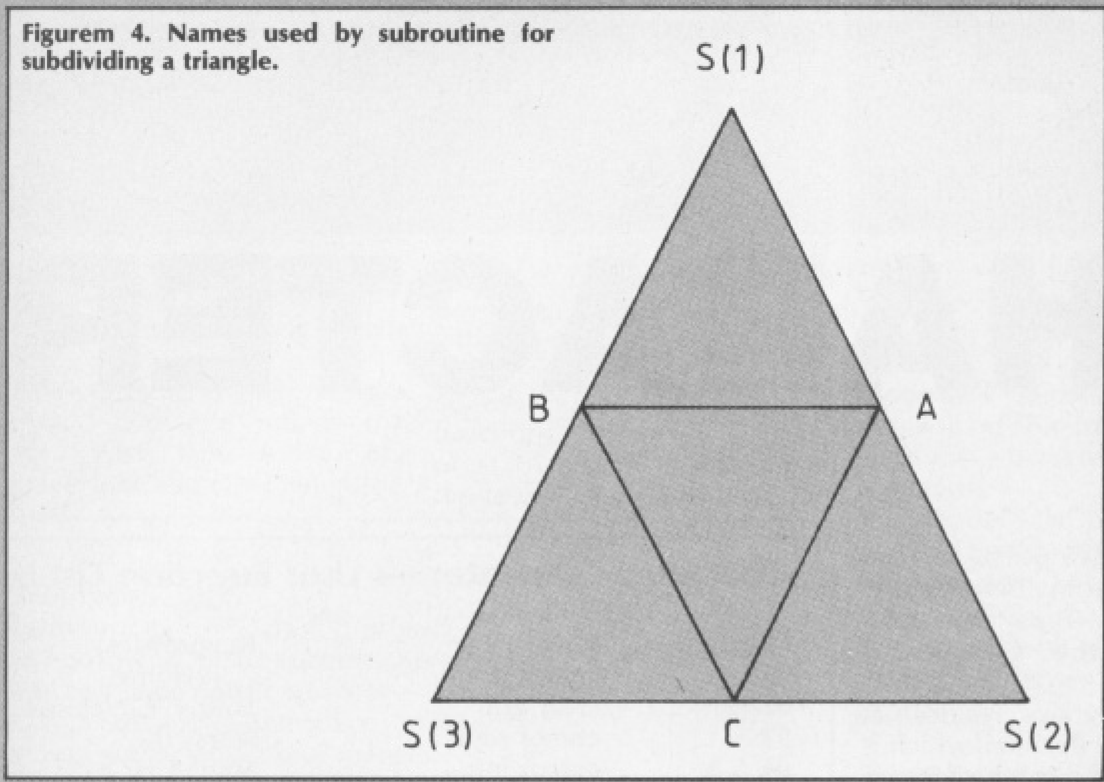


Figure 3. The use of arrays to interface the subroutines at lines 4000 and 5000. a) First sub-division b) Second sub-division.



Figurem 4. Names used by subroutine for subdividing a triangle.

of 4^N when N=1. (Although the program, as listed, goes no further than this, after the second sub-division each of four triangles will have been divided into four to give 4*4 or 4^2 triangles, and so on. In this way, the program is ready to be extended to create more detailed graphics).

Line 90 calls a subroutine starting at line 4000 to sub-divide the triangles that are repeatedly passed to it. This subroutine places the x andy co-ordinates of the corners of the triangle that is being dealt with in the arrays S and T. Then it calls the subroutine starting at line 5000 to do the actual sub-division. After it has done the dividing up, this subroutine leaves the x and y co-ordinates of the corners of the four triangles that result from the sub-division in the arryas U and V, respectively. The contents of these arrays are then transferred to the arrays XT and YT. This gives us the subroutine as:

```
4000 FOR M=1 TO 3
4010 S(M)=X(M+3*(K-1)): T(M)=Y(M+3*
(K-1))
4020 NEXT M
4030 GOSUB 5000: REM SUB-DIVIDE THIS
TRIANGLE
4040 FOR M=1 TO 12
4050 XT(M+12*(k-1))=U(M): YT(M+12*(K-
1))=V(M)
4060 NEXT M
4070 RETURN
```

The interface between the subroutines starting at 4000 and 5000 is illustrated in Figure 3, showing how they communicate by means of the arrays. The top illustration shows what happens with the initial sub-division. This does not exactly make it clear that all the arrays are needed, but the bottom illustration does so by showing how the arrays are used for the second sub-division.

Figure 4 shows how the x co-ordinates of the corners of a triangle are named during a sub-division as performed by the subroutine starting at line 5000. The subroutine itself is:

```
5000 A=0.5*(S(1)+S(2)
5010 D=0.5*(T(1)+T(2))
5020 B=0.5*(S(1)+S(3))
5030 E=0.5*(T(1)+(T(3))
5040 C=0.5*(S(2)+S(3))
5050 F=0.5*(T(2)+T(3))
5060 U(1)=S(1): U(4)=S(2): U(7)=S(3)
5070 V(1)=T(1); V(4)=T(2): V(7)=T(3)
5080 U(2)=A: U(6)=A: U(12)=A
5090 V(2)=D: V(6)=D: V(12)=D
5100 U(3)=B: U(8)=B: U(11)=B
5110 V(3)=E: V(8)=E: V(11)=E
5120 U(5)=C: U(9)=C: U(10)=C
5130 V(5)=F: V(9)=F: V(10)=F
5140 RETURN
```

After this, lines 110 to 130 copy copy the positions of the triangles fromXT and YT to X and Y so that they will be available for the next round of sub-dividing.

When the positions of all the triangles have been found they are plotted by lines 150 to 200. Here the inner loop from line 160 to 180, controlled by J, passes the co-ordinates of each triangle in turn to the arrays S and T so that the subroutine starting at line 3000 can be called to plot the triangle. The outer loop, controlled by K, ensures that all the triangles are dealt with.

The triangle-plotting routine simply calls our much used line-drawing subroutine (which begins at line 2000) three times. The line drawing routine draws a line from (X1, Y1) to (X2, Y2) and this gives the triangle-plotting routine as:

```
3000 X1=S(3): Y1=T(3)
3010 FOR P=1 TO 3
3020 X2=S(P): Y2=T(P)
3030 GOSUB 2000: REM DRAW LINE
3040 X1=X2: Y1=Y2
3050 NEXT P
3060 RETURN
```

Apart from the previously-used routines, this gives the complete program for creating the regular triangular mesh. It is now a simple matter to perturb the points at the corners of the triangles randomly before the triangles are plotted, so that we get our moutain. The perturbing can be done in the innermost subroutine, the one starting at line 5000, by adding to the co-ordinates a random amount with a variation that is proportional to the length of the side being bisected. For simplicity, the perturbation has been taken as a random amount between +/- 0.1*(length of the line in the x-direction). This gives the amended subroutine as:

```
5000 A1=(S(1)-S(2))*0.2*(RND(0)-0.5)
5010 B1=(S(1)-S(3))*0.2*(RND(0)-0.5)
5020 C1=(S(2)-S(3))*0.2*(RND(0)-0.5)
5030 A=0.5*(S(1)+S(2)) + A1
5040 D=0.5*(R(1)+T(2)) + A1
5050 B=0.5*(S(1)+S(3)) +B1
5060 E=0.5*(R(1)+T(3)) + B1
5070 C=0.5*(S(2)+S(3)) + C1
5080 F=0.5*(T(2)+T(3)) + C1
5090 U(1)=S(1): U(4)=S(2): U(7)=S(3)
5100 V(1)=T(1): V(4)=T(2): V(7)=T(3)
5110 U(2)=A: U(6)=A: U(12)=A
5120 V(2)=D: V(6)=D: V(12)=D
5130 U(3)=B: U(8)=B: U(11)=B
5140 V(3)=E: V(8)=E: V(11)=E
5150 U(5)=C: U(9)=C: U(10)=C
5160 V(5)=F: V(9)=: V(10)=F
5170 RETURN
```

The complete listing of the program to create the artificial landscape is given as Figure 5.

Further developments

The program can be extended to take the sub-division process further, so producing a more detailed landscape. The program only needs a few numerical changes for this which follow at once from the number of triangles there will be in the new arrangement. Don't forget to change the dimensions of the arrays! The way that the randomness is introduced can be made much more general, and the constant of proportionately for the range of the perturbations can be changed. You may like to experiment with this to see if you can characterise the differences between the types of landscapes that are produced by significantly different values for the constant of porportionality. Remember that because the perturbations are random, each rub of a program gives a different picture, although one that belongs to the same family.

IN CHARACTER

If you want to design your own games or use high-resolution and multi-colour characters, then this program by John McHale could be just what you're looking for.

Setting up

TYPE IN THE LISTING PROVIDED and save it on a disk or tape before attempting to run it.

Now RUN it and if all has gone smoothly, you will be given the option of saving the 4K Machine Code File to the device of your preference.

You should make it a habit, to 'verify' all programs that you save so as not to end up being disappointed, if you are not able to reload the program. It would also be a good idea to make a second copy, just in case you 'scratch' the first by accident.

For those of you who are not fortunate enough to enter the data correctly first time, you will be given one of two possible error reports (or maybe both).

These two error reports are as follows:

1 Insufficient/Too Many Data Items Error.
2 Checksum Error.

If you get error 1, then you will almost certainly get error 2 as well. If you get error 2 on its own, then you have entered some of the Data Items incorrectly e.g. you may have entered 250 instead of 240. Error 1 is self explanatory.

It would be useful to have a friend close by, to read out the data statements to you, in order to minimise the risk of errors.

Using the Program

Load the code by typing –

Load " ",1,1 or Lad "GEN'64.",1,1 for the tape version
Load "GEN'64.",8,1 for the disk version

Now type 'SYS 64738' to cold start the machine before using the program. Type 'SYS 52882' to enter the Characater Generator.

The Program resides at $C000 (49152 decimal) which is well out of the reach of Basic.

The user has the ability, using this versatile utility, to write a Basic program of up to 10K and at the same time, being able to enter and exit the Character Generator at will.

It is important to note that the data for your user defined graphics, resides in a 2K block – ($3000-$37FF) or 12288 to 14335 decimal and the complete set may be accessed by typing 'POKE 53272,28'.

Users who delight in designing games, using high-resolution and multi-colour characters, will find this program an invaluable aid for their graphics. A maximum of 256 Characters (which is more than enough) may be redefined. I have allocated these 256 Characters into four Subsets of 64 Characters each.

Alert users will notice, that when you enter the Character Generator for the first time, the standard PET character set is loaded into the user definable area, but this only happens once.

Try this exercise. Switch off the Machine. Load in the Character Generator. Type 'SYS 64738' and POKE 53272. You will not note that all the characters have turned to garbage. Type SYS 52882 and press 'Q' for quit. You should be back in Basic at this stage. Now type POKE 53272,28. Notice that the PET character set has now been loaded down. Type POKE 12288,255 and press ' ' and you will notice a straight line going across the top of the

Character Generator — User Function List

Function	Name	Keypress(es)
01.	cursor home	'Clr/Home'
02.	clear grid	Shift + 'Clr/Home'
03.	cursor right	'Csrs l/r'
04.	cursor left	Shift+'Crsr l/r'
05.	cursor down	'Crsr u/d'
06.	cursor up	Shift+ 'Crsr u/d'
07.	fill	'.'
08.	delete	'Del'
09.	space	'Spc'
10.	new line	'Ret'
11.	quick fill	'F'
12.	quick rub	'D'
13.	enable wrap mode	'W'
14.	disable wrap mode	'W'
15.	select character	' '
16.	set+	'+'
17.	set-	'-'
18.	next char	' '
19.	last char	'*'
20.	enable multicolour	'M'
21.	disable multicolour	'M'
22.	speed cursor	';'
23.	slow cursor	':'
24.	update cursor colour	'='
25.	update multicolour 1	'1'
26.	update multicolour 2	'2'
27.	update screen Colour	' '
28.	shift right	'F7'
29.	shift left	'F5'
30.	shift down	'F3'
31.	shift up	'F1'
32.	90° Rotate	'A' or 'C'
33.	invert character	'I'
34.	180° rotate	'R'
35.	transfer character	'T'
36.	transfer set	CTRL+'T'
37.	reverse video	'9'
38.	recall	'£'
39.	load characters	'L'
40.	save characters	'S'
41.	quit	'Q'

' ' sign. Now type SYS 52882 and you will notice that this character has not been changed.

The Program uses a very powerful Raster Interrupt, to enable the CBM set and the user defined graphics to be displayed at the same time. Their routine also handles such effects as split screen colour etc.

Reading the next section should convince you of the value of this extremely powerful Character Generator.

Select

When you first use the Character Generator, you will notice four lines to the right of the grid, which generate status reports. The most significant of these is the first, i.e. Character Mode. On running the program, this is set to 'Edit', which is the default mode. In this mode, you are able to access all of the listed functions. However, if you press ' ', then the character mode will change to select.

Immediately, you will see that the cursor is no longer on the grid, but is now flashing on a line of characters which are directly below the grid. The only keys valid in this mode are Shift, Crsr l/r ' 'E'.

Use the Crsr l/r key in the normal way (in conjunction with the Shift key) to locate the character you wish to edit. Now press 'E' and the character mode will return to 'Edit' and you will be able to experiment with the character you have selected.

Alternatively, as mentioned in the Function List, you may use functions 18 and 19 (Next Char & Last Char) to achieve the same result without ever having to enter the Select Mode.

Transfer Set

To transfer a set of 64 characters to another location, first press CTRL & 'T'. You should now see the Character Set no. (fourth status line) flashing.

Using the functions 'Set +' and 'Set -', choose the set that you want the original to be transferred to.

Now press 'Return', the transfer will be completed and normal operation restored.

Recall

Have you ever made a complete mess of a sprite or UDG that you were designing and wished you could restore it to its original form. This is what 'Recall' is for.

Every character you use is automatically 'buffered' so that in the event of you using the functions excessively, pressing 'Recall' will restore the character to its original form.

A Finishing Note

Do not try to define your first blockbusting character set immediately. Mess around with the various functions, until you become familiar with them. I hope you enjoy using it.

Description

Positions cursor in top left of grid.
As above and clears current character.
Moves cursor right one space.
Moves cursor left one space.
Moves cursor down one space.
Moves cursor up one space.
Fills space at current cursor location.
Deletes space to left of cursor.
Rubs space to right of cursor.
Places cursor on left of next line down.
Fills left-right on current line.
Deletes left-right on current line.
Allows code to move off the edge of the grid and appear on the opposite side
Keeps cursor within grid boundaries.
Enables user to select the next character to be edited.
Advances to next subset of 64 chars.
Returns to previous subset.
Allows the user to move to the next character without having to enter 'Select Mode' – see notes on 15.
As above but moves backwards rather than forwards.
Enables character multicolour mode.
Disables character multicolour mode.
Speeds cursor's response.
Slows cursor response
Self explanatory.
Self explanatory
Self explanatory
Self explanatory (Split screen colour)
Moves complete character right (1 Bit).
Moves complete character left (1 bit).
Moves complete character down (1 bit).
Moves complete character up (1 bit).
Rotate character through 90 degrees: 'A' – anticlockwise 'C' clockwise
Flips character upside down.
Creates Mirror image of character through the vertical axis. (Works in hi/res & multicolour modes).
Copies one character in the current set into another character in the same set.
Transfers one complete set to another.
'9'=Rvs Char: Ctrl+'9 = Rvs Set.
Recalls 'buffered' character (See special notes on this function).
Load character sets from tape or disk.
Saves character sets to tape or disk. (note: pressing 'S' will save the current subset (64 chars.); pressing 'Ctrl'+'S' will save the entire four sets.
Exit to Basic.

Program Listing

```
0 REM  :::::::::::::::::::::::::
1 REM  :                       :
2 REM  : CHARACTER GENERATOR   :
3 REM  :                       :
4 REM  : 4K PURE M/CODE        :
5 REM  :                       :
6 REM  : OVER 40 FUNCTIONS.    :
7 REM  :                       :
8 REM  : WRITTEN BY J.MC HALE  :
9 REM  :                       :
10 REM : STRANDHILL RD.,SLIGO, :
11 REM :                       :
12 REM : REP. IRELAND.         :
13 REM :                       :
14 REM : DEDICATED TO:         :
15 REM :                       :
16 REM : JENNIFER .........    :
17 REM :                       :
18 REM :::::::::::::::::::::::::
```

Program Listing (cont.)

```
20 PRINT"[CLEAR]":REM CLR/HOME
30 POKE53281,0:POKE53280,0
40 PRINT"[YELLOW]PLEASE WAIT, WRIT
ING CODE TO MEMORY ."
50 SA=49152:BC=0:TL=0
60 READA:IFA=-1THEN80
70 POKESA+BC,A:BC=BC+1:TL=TL+A:GOT
O60
80 REM * ERROR TRAPPING *
90 IFBC<>3882THEN400
100 IFTL<>483826THEN400
110 POKE53280,14:POKE53281,6:PRINT
"[CLEAR][c 7]":REM CLR/HOME & LIGH
T BLUE ( CTR
L+'7')
120 PRINT"[DOWN]OKAY - CODE ENTERE
D CORRECTLY ."
130 PRINT"[DOWN]YOU MAY NOW SAVE T
HE PROGRAM CODE TO     "
140 PRINT"EITHER TAPE OR DISK."
150 DV=1
160 INPUT"[DOWN]WHICH DEVICE ( T O
R D ) :  ";D$
170 IFD$<>"D"ANDD$<>"T"THEN160
180 IFD$="D"THENDV=8
190 IFDV=8THEN220
200 PRINT"[DOWN]PLACE [RVSON]BLANK
[RVSOFF] CASSETTE IN C2N UNIT, REW
IND"
210 PRINT"AND PRESS STOP/EJECT.":G
OTO230
220 PRINT"PLACE [RVSON]FORMATTED[R
VSOFF] DISK IN DRIVE#0 (DEV 8)."
230 PRINT"[DOWN][DOWN]PRESS '*' WH
EN READY .[DOWN][DOWN]"
240 POKE198,0
250 GETA$:IFA$<>"*"THEN250
260 POKE198,0:POKE2,DV:SYS52992
300 PRINT"[DOWN]PLEASE VERIFY CODE
 BY TYPING :"
310 PRINT"[DOWN]TAPE VERSION - 'VE
RIFY"CHR$(34)CHR$(34)",1,1'"
320 PRINT"[DOWN]DISK VERSION - 'VE
RIFY"CHR$(34)"GEN'64."CHR$(34)",8,
1'"
330 PRINT"[DOWN]THIS IS A PRECAUTI
ONARY MEASURE TO MAKE "
340 PRINT"SURE THAT THE CODE HAS B
EEN SAVED        "
350 PRINT" CORRECTLY."
360 END
400 REM * ANALYSE ERRORS AND REPOR
T *
410 IFTL<>483826THENPRINT"[DOWN]CH
ECKSUM ERROR."
420 IFBC=3882THEN460
430 IFBC<3882THENPRINT"[DOWN]INSUF
FICIENT ";:GOTO450
440 PRINT"[DOWN]TOO MANY ";
450 PRINT"DATA ITEMS."
460 PRINT"[DOWN][DOWN]CHECK DATA S
TATEMENTS CAREFULLY.":STOP
1000 DATA  160, 151, 146, 137, 148
, 148, 133, 142, 160, 130
1010 DATA  153, 160, 160, 186, 160
, 160, 138, 143, 136, 142
1020 DATA  160, 141, 131, 160, 136
, 129, 140, 133, 174, 160
1030 DATA  168, 131, 169, 160, 160
, 147, 133, 144, 148, 133
1040 DATA  141, 130, 133, 146, 160
, 160, 177, 185, 184, 180
1050 DATA  160, 174, 160, 160, 160
, 160, 160, 160, 131, 136
1060 DATA  129, 146, 129, 131, 148
, 133, 146, 160, 141, 143
1070 DATA  132, 133, 160, 186, 141
, 173, 131, 143, 140, 143
1080 DATA  149, 146, 160, 160, 141
, 143, 132, 133, 160, 186
1090 DATA  131, 136, 129, 146, 129
, 131, 148, 133, 146, 160
1100 DATA  160, 147, 133, 148, 160
, 186, 133, 132, 137, 148
1110 DATA  160, 160, 147, 133, 140
, 133, 131, 148, 143, 134
1120 DATA  134, 143, 142, 160, 169
, 147, 32, 210, 255, 169
1130 DATA  6, 141, 33, 208, 73, 8,
 141, 32, 208, 162
1140 DATA  40, 169, 160, 157, 87,
6, 169, 7, 157, 87
1150 DATA  218, 202, 208, 243, 189
, 0, 192, 157, 11, 4
1160 DATA  189, 29, 192, 157, 91,
4, 169, 1, 157, 11
1170 DATA  216, 169, 3, 157, 91, 2
```

Program Listing (cont.)

```
16, 169, 160, 157, 171
1180 DATA  4, 157, 211, 4, 157, 25
1, 4, 157, 35, 5
1190 DATA  157, 75, 5, 157, 171, 2
16, 73, 1, 157, 75
1200 DATA  217, 157, 211, 216, 157
, 251, 216, 157, 35, 217
1210 DATA  232, 224, 29, 208, 195,
 169, 0, 133, 251, 133
1220 DATA  253, 170, 169, 4, 133,
252, 169, 216, 133, 254
1230 DATA  160, 0, 169, 160, 145,
251, 169, 4, 145, 253
1240 DATA  200, 192, 11, 208, 243,
 165, 251, 24, 105, 40
1250 DATA  133, 251, 133, 253, 165
, 252, 105, 0, 133, 252
1260 DATA  105, 212, 133, 254, 232
, 224, 11, 208, 217, 169
1270 DATA  72, 133, 251, 169, 4, 1
33, 252, 162, 176, 160
1280 DATA  9, 138, 145, 251, 153,
40, 4, 165, 251, 24
1290 DATA  105, 41, 133, 251, 165,
 252, 105, 0, 133, 252
1300 DATA  136, 232, 224, 184, 208
, 231, 162, 0, 189, 58
1310 DATA  192, 157, 211, 4, 189,
74, 192, 157, 251, 4
1320 DATA  189, 198, 193, 157, 35,
 5, 232, 224, 16, 208
1330 DATA  233, 96, 120, 173, 14,
220, 41, 254, 141, 14
1340 DATA  220, 173, 17, 208, 41,
127, 141, 17, 208, 169
1350 DATA  132, 141, 20, 3, 169, 1
93, 141, 21, 3, 169
1360 DATA  177, 141, 18, 208, 173,
 26, 208, 9, 1, 141
1370 DATA  26, 208, 88, 96, 173, 2
5, 208, 9, 1, 141
1380 DATA  25, 208, 104, 168, 104,
 170, 104, 64, 169, 1
1390 DATA  44, 25, 208, 240, 243,
173, 24, 208, 41, 8
1400 DATA  208, 40, 160, 28, 174,
18, 207, 169, 49, 141
1410 DATA  18, 208, 173, 14, 207,
208, 8, 173, 22, 208
1420 DATA  41, 239, 76, 174, 193,
173, 22, 208, 9, 16
1430 DATA  141, 22, 208, 142, 33,
208, 140, 24, 208, 76
1440 DATA  118, 193, 162, 6, 160,
20, 169, 177, 141, 18
1450 DATA  208, 76, 161, 193, 151,
 146, 129, 144, 129, 146
1460 DATA  143, 149, 142, 132, 160
, 160, 160, 160, 160, 186
1470 DATA  133, 142, 129, 130, 140
, 133, 132, 160, 132, 137
1480 DATA  147, 129, 130, 140, 133
, 132, 120, 169, 49, 141
1490 DATA  20, 3, 169, 234, 141, 2
1, 3, 173, 14, 220
1500 DATA  9, 1, 141, 14, 220, 88,
 96, 0, 0, 0
1510 DATA  173, 14, 220, 41, 254,
141, 14, 220, 165, 1
1520 DATA  41, 251, 133, 1, 169, 0
, 133, 251, 133, 253
1530 DATA  169, 208, 133, 252, 169
, 48, 133, 254, 162, 0
1540 DATA  160, 0, 177, 251, 145,
253, 200, 208, 249, 230
1550 DATA  252, 230, 254, 232, 224
, 8, 208, 238, 165, 1
1560 DATA  9, 4, 133, 1, 173, 14,
220, 9, 1, 141
1570 DATA  14, 220, 234, 96, 169,
0, 133, 254, 162, 8
1580 DATA  10, 38, 254, 6, 252, 14
4, 7, 24, 101, 251
1590 DATA  144, 2, 230, 254, 202,
208, 239, 133, 253, 96
1600 DATA  173, 12, 207, 133, 251,
 169, 8, 133, 252, 32
1610 DATA  62, 194, 165, 254, 24,
105, 48, 133, 254, 169
1620 DATA  82, 133, 249, 133, 251,
 169, 4, 133, 250, 169
1630 DATA  216, 133, 252, 160, 0,
177, 253, 170, 152, 72
1640 DATA  138, 162, 8, 160, 0, 10
, 72, 144, 8, 169
1650 DATA  81, 145, 249, 145, 251,
 208, 8, 169, 43, 145
1660 DATA  249, 169, 0, 145, 251,
104, 200, 202, 208, 231
1670 DATA  165, 249, 24, 105, 40,
133, 249, 133, 251, 165
1680 DATA  250, 105, 0, 133, 250,
105, 212, 133, 252, 104
1690 DATA  168, 200, 192, 8, 208,
195, 234, 96, 169, 64
1700 DATA  133, 251, 173, 13, 207,
```

Program Listing (cont.)

```
 133, 252, 32, 62, 194
1710 DATA  162, 0, 160, 4, 169, 1,
 153, 224, 217, 153
1720 DATA  8, 218, 138, 24, 101, 2
53, 153, 224, 5, 24
1730 DATA  105, 32, 153, 8, 6, 232
, 200, 192, 36, 208
1740 DATA  229, 169, 188, 133, 248
, 133, 250, 169, 6, 133
1750 DATA  249, 169, 218, 133, 251
, 169, 0, 133, 2, 162
1760 DATA  0, 160, 0, 138, 24, 101
, 253, 145, 248, 173
1770 DATA  19, 207, 72, 173, 14, 2
07, 240, 5, 104, 9
1780 DATA  8, 208, 1, 104, 145, 25
0, 232, 200, 192, 16
1790 DATA  208, 227, 165, 248, 24,
 105, 80, 133, 248, 133
1800 DATA  250, 165, 249, 105, 0,
133, 249, 105, 212, 133
1810 DATA  251, 230, 2, 165, 2, 20
1, 4, 208, 198, 96
1820 DATA  169, 169, 133, 250, 133
, 252, 169, 6, 133, 251
1830 DATA  169, 218, 133, 253, 169
, 0, 133, 2, 170, 160
1840 DATA  0, 173, 12, 207, 145, 2
50, 138, 145, 252, 232
1850 DATA  200, 200, 192, 8, 208,
241, 165, 250, 24, 105
1860 DATA  80, 133, 250, 133, 252,
 165, 251, 105, 0, 133
1870 DATA  251, 105, 212, 133, 253
, 230, 2, 165, 2, 201
1880 DATA  4, 208, 212, 96, 173, 4
, 207, 133, 250, 173
1890 DATA  5, 207, 133, 251, 172,
7, 207, 177, 250, 41
1900 DATA  63, 141, 12, 207, 169,
64, 133, 251, 173, 13
1910 DATA  207, 133, 252, 32, 62,
194, 173, 12, 207, 24
1920 DATA  101, 253, 141, 12, 207,
 96, 173, 8, 207, 133
1930 DATA  250, 173, 9, 207, 133,
251, 172, 10, 207, 96
1940 DATA  32, 158, 195, 160, 0, 1
69, 0, 133, 2, 177
1950 DATA  250, 72, 74, 74, 74, 74
, 74, 74, 24, 101
1960 DATA  2, 133, 2, 104, 72, 10,
 10, 10, 10, 10
1970 DATA  10, 24, 101, 2, 133, 2,
 104, 72, 41, 48
1980 DATA  74, 74, 24, 101, 2, 133
, 2, 104, 41, 12
1990 DATA  10, 10, 24, 101, 2, 145
, 250, 200, 192, 8
2000 DATA  208, 199, 96, 32, 158,
195, 160, 0, 169, 0
2010 DATA  133, 2, 162, 8, 177, 25
0, 10, 72, 144, 9
2020 DATA  165, 2, 74, 9, 128, 133
, 2, 208, 2, 70
2030 DATA  2, 104, 202, 208, 237,
165, 2, 145, 250, 200
2040 DATA  192, 8, 208, 220, 96, 3
2, 158, 195, 169, 7
2050 DATA  133, 2, 160, 0, 152, 72
, 177, 250, 170, 164
2060 DATA  2, 177, 250, 72, 138, 1
45, 250, 104, 170, 104
2070 DATA  168, 138, 145, 250, 198
, 2, 200, 192, 4, 208
2080 DATA  229, 96, 152, 72, 177,
250, 164, 2, 145, 250
2090 DATA  104, 168, 96, 32, 158,
195, 169, 7, 133, 2
2100 DATA  160, 6, 32, 58, 196, 16
9, 0, 145, 250, 198
2110 DATA  2, 136, 192, 255, 208,
242, 96, 32, 158, 195
2120 DATA  162, 0, 134, 2, 160, 1,
 32, 58, 196, 230
2130 DATA  2, 200, 192, 8, 208, 24
6, 136, 138, 145, 250
2140 DATA  96, 32, 158, 195, 160,
0, 177, 250, 10, 145
2150 DATA  250, 200, 192, 8, 208,
246, 96, 32, 158, 195
2160 DATA  169, 128, 133, 2, 160,
8, 169, 0, 153, 247
2170 DATA  207, 136, 208, 250, 152
, 72, 177, 250, 160, 0
2180 DATA  74, 144, 10, 72, 185, 2
48, 207, 5, 2, 153
2190 DATA  248, 207, 104, 200, 192
, 8, 208, 238, 70, 2
2200 DATA  104, 168, 200, 192, 8,
208, 223, 160, 0, 185
2210 DATA  248, 207, 145, 250, 200
, 192, 8, 208, 246, 96
2220 DATA  238, 18, 207, 96, 238,
34, 208, 96, 238, 35
2230 DATA  208, 96, 238, 39, 208,
```

Program Listing (cont.)

```
96, 238, 19, 207, 96
2240 DATA  173, 17, 207, 56, 233,
1, 208, 1, 96, 141
2250 DATA  17, 207, 96, 173, 17, 2
07, 24, 105, 1, 144
2260 DATA  244, 96, 173, 14, 207,
73, 1, 141, 14, 207
2270 DATA  32, 101, 199, 96, 172,
3, 207, 136, 208, 21
2280 DATA  173, 23, 207, 240, 15,
169, 1, 141, 11, 207
2290 DATA  169, 96, 141, 0, 208, 1
69, 8, 141, 3, 207
2300 DATA  96, 140, 3, 207, 14, 11
, 207, 173, 0, 208
2310 DATA  56, 233, 8, 141, 0, 208
, 96, 172, 3, 207
2320 DATA  200, 192, 9, 208, 21, 1
73, 23, 207, 240, 15
2330 DATA  169, 128, 141, 11, 207,
169, 40, 141, 0, 208
2340 DATA  169, 1, 141, 3, 207, 96
, 140, 3, 207, 78
2350 DATA  11, 207, 173, 0, 208, 2
4, 105, 8, 141, 0
2360 DATA  208, 96, 174, 2, 207, 2
02, 208, 30, 173, 23
2370 DATA  207, 240, 24, 169, 105,
141, 0, 207, 169, 5
2380 DATA  141, 1, 207, 162, 8, 14
2, 2, 207, 202, 142
2390 DATA  10, 207, 169, 122, 141,
1, 208, 96, 142, 2
2400 DATA  207, 206, 10, 207, 173,
0, 207, 56, 233, 40
2410 DATA  141, 0, 207, 173, 1, 20
7, 233, 0, 141, 1
2420 DATA  207, 173, 1, 208, 56, 2
33, 8, 141, 1, 208
2430 DATA  96, 174, 2, 207, 232, 2
24, 9, 208, 30, 173
2440 DATA  23, 207, 240, 24, 169,
81, 141, 0, 207, 169
2450 DATA  4, 141, 1, 207, 162, 1,
142, 2, 207, 202
2460 DATA  142, 10, 207, 169, 66,
141, 1, 208, 96, 142
2470 DATA  2, 207, 238, 10, 207, 1
73, 0, 207, 24, 105
2480 DATA  40, 141, 0, 207, 173, 1
, 207, 105, 0, 141
2490 DATA  1, 207, 173, 1, 208, 24
, 105, 8, 141, 1
2500 DATA  208, 96, 173, 141, 2, 4
1, 1, 208, 4, 32
2510 DATA  37, 197, 96, 32, 250, 1
96, 96, 173, 141, 2
2520 DATA  41, 1, 208, 4, 32, 151,
197, 96, 32, 82
2530 DATA  197, 96, 162, 63, 169,
0, 157, 64, 3, 202
2540 DATA  208, 250, 73, 255, 157,
64, 3, 232, 232, 232
2550 DATA  224, 24, 208, 246, 96,
169, 128, 162, 74, 160
2560 DATA  70, 141, 137, 196, 142,
156, 196, 140, 174, 196
2570 DATA  32, 133, 196, 96, 169,
1, 162, 10, 160, 6
2580 DATA  76, 25, 198, 173, 0, 20
7, 133, 250, 173, 1
2590 DATA  207, 133, 251, 172, 3,
207, 96, 32, 158, 195
2600 DATA  160, 0, 177, 250, 73, 2
55, 145, 250, 200, 192
2610 DATA  8, 208, 245, 96, 169, 0
, 133, 250, 173, 13
2620 DATA  207, 10, 24, 105, 48, 1
33, 251, 162, 0, 160
2630 DATA  0, 177, 250, 73, 255, 1
45, 250, 200, 208, 247
2640 DATA  230, 251, 232, 224, 2,
208, 238, 96, 173, 141
2650 DATA  2, 41, 4, 208, 4, 32, 6
1, 198, 96, 32
2660 DATA  78, 198, 96, 173, 21, 2
08, 73, 1, 141, 21
2670 DATA  208, 96, 173, 21, 208,
73, 2, 141, 21, 208
2680 DATA  96, 169, 10, 141, 124,
196, 32, 117, 196, 96
2690 DATA  169, 74, 208, 245, 32,
212, 196, 76, 186, 194
2700 DATA  173, 4, 207, 133, 250,
173, 5, 207, 133, 251
2710 DATA  172, 7, 207, 96, 32, 15
8, 195, 169, 0, 145
2720 DATA  250, 96, 32, 158, 195,
169, 255, 145, 250, 96
2730 DATA  173, 14, 207, 208, 4, 3
2, 235, 195, 96, 32
2740 DATA  172, 195, 96, 169, 66,
141, 1, 208, 169, 40
2750 DATA  141, 0, 208, 169, 81, 1
41, 0, 207, 169, 4
2760 DATA  141, 1, 207, 162, 1, 14
```

```
2, 2, 207, 142, 3
2770 DATA  207, 202, 142, 10, 207,
 169, 128, 141, 11, 207
2780 DATA  173, 141, 2, 41, 1, 208
, 1, 96, 32, 158
2790 DATA  195, 169, 0, 145, 250,
200, 192, 8, 208, 249
2800 DATA  96, 162, 0, 189, 90, 19
2, 157, 75, 5, 232
2810 DATA  224, 16, 208, 245, 96,
173, 13, 207, 24, 105
2820 DATA  176, 141, 92, 5, 162, 0
, 173, 15, 207, 208
2830 DATA  5, 189, 106, 192, 208,
3, 189, 112, 192, 157
2840 DATA  228, 4, 232, 224, 6, 20
8, 235, 173, 23, 207
2850 DATA  208, 5, 189, 216, 193,
208, 3, 189, 208, 193
2860 DATA  157, 46, 5, 232, 224, 1
4, 208, 235, 173, 14
2870 DATA  207, 208, 5, 189, 104,
192, 208, 3, 189, 107
2880 DATA  192, 157, 254, 4, 232,
224, 17, 208, 235, 32
2890 DATA  9, 199, 96, 32, 23, 199
, 206, 19, 207, 32
2900 DATA  158, 198, 96, 173, 23,
207, 73, 1, 141, 23
2910 DATA  207, 32, 23, 199, 96, 1
73, 12, 207, 133, 251
2920 DATA  169, 8, 133, 252, 32, 6
2, 194, 165, 253, 141
2930 DATA  8, 207, 165, 254, 24, 1
05, 48, 141, 9, 207
2940 DATA  96, 32, 186, 194, 32, 2
3, 199, 32, 116, 195
2950 DATA  32, 123, 199, 32, 88, 1
94, 32, 52, 195, 96
2960 DATA  173, 13, 207, 24, 105,
1, 201, 4, 208, 1
2970 DATA  96, 141, 13, 207, 76, 5
3, 202, 173, 13, 207
2980 DATA  56, 233, 1, 201, 255, 2
08, 240, 96, 173, 21
2990 DATA  208, 41, 1, 141, 21, 20
8, 96, 173, 21, 208
3000 DATA  9, 2, 141, 21, 208, 96,
 172, 7, 207, 200
3010 DATA  192, 33, 208, 90, 174,
6, 207, 232, 224, 3
3020 DATA  208, 46, 173, 23, 207,
208, 1, 96, 32, 196
3030 DATA  199, 169, 227, 141, 4,
207, 169, 5, 141, 5
3040 DATA  207, 169, 146, 141, 3,
208, 169, 1, 141, 6
3050 DATA  207, 141, 7, 207, 169,
56, 141, 2, 208, 169
3060 DATA  0, 141, 16, 208, 32, 20
5, 199, 96, 142, 6
3070 DATA  207, 32, 196, 199, 173,
 4, 207, 24, 105, 40
3080 DATA  141, 4, 207, 173, 5, 20
7, 105, 0, 141, 5
3090 DATA  207, 173, 3, 208, 24, 1
05, 8, 141, 3, 208
3100 DATA  169, 1, 208, 203, 140,
7, 207, 173, 2, 208
3110 DATA  24, 105, 8, 72, 144, 8,
 173, 16, 208, 9
3120 DATA  2, 141, 16, 208, 104, 1
41, 2, 208, 96, 172
3130 DATA  7, 207, 136, 208, 88, 1
74, 6, 207, 202, 208
3140 DATA  48, 173, 23, 207, 208,
1, 96, 32, 196, 199
3150 DATA  169, 11, 141, 4, 207, 1
69, 6, 141, 5, 207
3160 DATA  169, 154, 141, 3, 208,
169, 2, 141, 6, 207
3170 DATA  169, 32, 141, 7, 207, 1
69, 48, 141, 2, 208
3180 DATA  169, 2, 141, 16, 208, 3
2, 205, 199, 96, 142
3190 DATA  6, 207, 32, 196, 199, 1
73, 4, 207, 56, 233
3200 DATA  40, 141, 4, 207, 173, 5
, 207, 233, 0, 141
3210 DATA  5, 207, 173, 3, 208, 56
, 233, 8, 141, 3
3220 DATA  208, 208, 203, 140, 7,
207, 173, 2, 208, 56
3230 DATA  233, 8, 72, 176, 5, 169
, 0, 141, 16, 208
3240 DATA  104, 141, 2, 208, 96, 1
69, 1, 141, 15, 207
3250 DATA  32, 23, 199, 32, 159, 2
55, 165, 197, 201, 2
3260 DATA  208, 28, 173, 141, 2, 4
1, 1, 208, 6, 32
3270 DATA  214, 199, 76, 230, 200,
 32, 81, 200, 32, 2
3280 DATA  201, 32, 136, 198, 32,
155, 199, 76, 205, 200
3290 DATA  201, 14, 208, 240, 169,
```

Program Listing (cont.)

```
0, 141, 15, 207, 32
3300 DATA  23, 199, 32, 14, 201, 9
6, 172, 17, 207, 162
3310 DATA  0, 202, 208, 253, 136,
208, 248, 96, 32, 158
3320 DATA  195, 160, 0, 177, 250,
153, 240, 207, 200, 192
3330 DATA  8, 208, 246, 96, 32, 15
8, 195, 160, 0, 185
3340 DATA  240, 207, 145, 250, 200
, 192, 8, 208, 246, 96
3350 DATA  32, 214, 199, 76, 47, 2
02, 32, 81, 200, 76
3360 DATA  47, 202, 32, 158, 195,
173, 11, 207, 73, 255
3370 DATA  49, 250, 145, 250, 96,
32, 158, 195, 177, 250
3380 DATA  13, 11, 207, 145, 250,
32, 37, 197, 76, 88
3390 DATA  194, 32, 250, 196, 32,
58, 201, 76, 88, 194
3400 DATA  32, 58, 201, 32, 37, 19
7, 76, 88, 194, 32
3410 DATA  50, 197, 76, 151, 197,
0, 51, 2, 7, 60
3420 DATA  44, 0, 1, 21, 18, 50, 4
5, 53, 48, 46
3430 DATA  49, 54, 40, 43, 17, 33,
10, 20, 6, 3
3440 DATA  4, 5, 32, 57, 36, 56, 5
9, 8, 9, 62
3450 DATA  42, 13, 22, 0, 0, 0, 20
7, 222, 237, 96
3460 DATA  71, 87, 105, 186, 178,
216, 229, 208, 30, 104
3470 DATA  52, 46, 168, 185, 194,
21, 19, 38, 145, 154
3480 DATA  93, 69, 112, 196, 238,
200, 204, 158, 111, 170
3490 DATA  58, 25, 91, 0, 0, 0, 19
8, 197, 197, 201
3500 DATA  201, 201, 201, 198, 198
, 196, 196, 196, 201, 205
3510 DATA  201, 201, 199, 199, 198
, 196, 198, 198, 198, 198
3520 DATA  196, 196, 198, 196, 196
, 196, 196, 198, 199, 202
3530 DATA  206, 206, 205, 0, 0, 0,
32, 159, 255, 162
3540 DATA  0, 165, 197, 221, 112,
201, 208, 15, 189, 152
3550 DATA  201, 141, 46, 3, 189, 1
92, 201, 141, 47, 3
3560 DATA  108, 46, 3, 232, 224, 3
7, 208, 231, 96, 81
3570 DATA  4, 1, 1, 227, 5, 1, 1,
0, 48, 0
3580 DATA  128, 0, 0, 0, 0, 1, 40,
10, 0, 0
3590 DATA  1, 0, 1, 162, 0, 189, 9
, 202, 157, 0
3600 DATA  207, 232, 224, 24, 208,
245, 96, 32, 155, 199
3610 DATA  76, 14, 201, 32, 149, 1
99, 76, 14, 201, 169
3620 DATA  0, 32, 210, 255, 162, 0
, 32, 207, 255, 201
3630 DATA  13, 240, 10, 157, 32, 2
07, 232, 224, 16, 208
3640 DATA  241, 169, 13, 142, 22,
207, 32, 210, 255, 96
3650 DATA  169, 0, 141, 26, 208, 1
41, 21, 208, 169, 6
3660 DATA  141, 33, 208, 73, 8, 14
1, 32, 208, 169, 147
3670 DATA  32, 210, 255, 96, 169,
28, 141, 24, 208, 162
3680 DATA  0, 138, 157, 0, 4, 169,
1, 157, 0, 216
3690 DATA  232, 208, 244, 162, 0,
189, 149, 202, 32, 210
3700 DATA  255, 232, 224, 21, 208,
245, 108, 2, 3, 17
3710 DATA  17, 17, 17, 17, 17, 17,
13, 81, 85, 73
3720 DATA  84, 46, 13, 82, 69, 65,
68, 89, 46, 13
3730 DATA  32, 90, 202, 76, 125, 2
05, 12, 15, 1, 4
3740 DATA  46, 19, 1, 22, 5, 46, 1
73, 14, 220, 41
3750 DATA  254, 141, 14, 220, 165,
1, 41, 251, 133, 1
3760 DATA  169, 48, 133, 247, 169,
207, 133, 248, 162, 0
3770 DATA  138, 72, 189, 176, 202,
133, 251, 169, 8, 133
3780 DATA  252, 234, 32, 62, 194,
165, 254, 24, 105, 208
3790 DATA  133, 254, 160, 0, 177,
253, 145, 247, 200, 192
3800 DATA  8, 208, 247, 165, 247,
24, 105, 8, 133, 247
3810 DATA  165, 248, 105, 0, 133,
248, 104, 170, 232, 224
3820 DATA  10, 208, 203, 165, 1, 9
```

```
, 4, 133, 1, 173
3830 DATA  14, 220, 9, 1, 141, 14,
 220, 96, 173, 13
3840 DATA  207, 133, 255, 32, 159,
 255, 165, 197, 201, 1
3850 DATA  240, 31, 201, 40, 208,
6, 32, 168, 199, 76
3860 DATA  53, 203, 201, 43, 208,
3, 32, 185, 199, 173
3870 DATA  92, 5, 73, 128, 141, 92
, 5, 32, 2, 201
3880 DATA  76, 27, 203, 173, 13, 2
07, 10, 24, 105, 48
3890 DATA  133, 253, 169, 0, 133,
250, 133, 252, 165, 255
3900 DATA  10, 24, 105, 48, 133, 2
51, 162, 0, 160, 0
3910 DATA  177, 250, 145, 252, 200
, 208, 249, 230, 251, 230
3920 DATA  253, 232, 224, 2, 208,
238, 165, 255, 141, 13
3930 DATA  207, 173, 92, 5, 9, 128
, 141, 92, 5, 76
3940 DATA  53, 202, 169, 80, 133,
247, 133, 249, 133, 251
3950 DATA  169, 4, 133, 248, 133,
250, 162, 0, 138, 72
3960 DATA  160, 0, 152, 72, 177, 2
53, 160, 0, 10, 144
3970 DATA  17, 72, 169, 160, 145,
249, 165, 250, 24, 105
3980 DATA  212, 133, 252, 169, 1,
145, 251, 104, 200, 192
3990 DATA  8, 208, 231, 165, 249,
24, 105, 40, 133, 249
4000 DATA  133, 251, 165, 250, 105
, 0, 133, 250, 104, 168
4010 DATA  200, 192, 8, 208, 203,
165, 247, 24, 105, 8
4020 DATA  133, 247, 133, 249, 133
, 251, 165, 248, 105, 0
4030 DATA  133, 248, 133, 250, 165
, 253, 24, 105, 8, 133
4040 DATA  253, 165, 254, 105, 0,
133, 254, 104, 170, 232
4050 DATA  224, 5, 208, 160, 96, 1
69, 207, 133, 254, 173
4060 DATA  20, 207, 208, 4, 169, 4
8, 208, 2, 169, 88
4070 DATA  133, 253, 76, 126, 203,
 169, 0, 133, 250, 133
4080 DATA  252, 165, 170, 240, 6,
162, 48, 169, 56, 208
4090 DATA  8, 173, 13, 207, 32, 37
, 205, 234, 234, 133
4100 DATA  253, 134, 251, 96, 32,
32, 39, 17, 39, 32
4110 DATA  61, 32, 17, 21, 9, 20,
32, 59, 32, 39
4120 DATA  49, 39, 32, 61, 32, 20,
 1, 16, 5, 32
4130 DATA  59, 32, 39, 56, 39, 32,
 61, 32, 4, 9
4140 DATA  19, 11, 32, 32, 162, 40
, 189, 31, 204, 157
4150 DATA  183, 5, 169, 7, 157, 18
3, 217, 202, 208, 242
4160 DATA  96, 32, 2, 201, 162, 40
, 189, 183, 5, 73
4170 DATA  128, 157, 183, 5, 202,
208, 245, 96, 0, 29
4180 DATA  29, 29, 69, 78, 84, 69,
82, 32, 70, 73
4190 DATA  76, 69, 78, 65, 77, 69,
32, 65, 78, 68
4200 DATA  32, 80, 82, 69, 83, 83,
32, 39, 82, 69
4210 DATA  84, 85, 82, 78, 39, 13,
13, 29, 29, 29
4220 DATA  0, 70, 73, 76, 69, 78,
65, 77, 69, 32
4230 DATA  58, 32, 162, 13, 138, 3
2, 210, 255, 202, 208
4240 DATA  250, 189, 106, 204, 32,
 210, 255, 232, 224, 54
4250 DATA  208, 245, 76, 59, 202,
32, 115, 205, 169, 0
4260 DATA  141, 32, 208, 141, 33,
208, 32, 239, 203, 32
4270 DATA  160, 204, 32, 72, 204,
234, 234, 234, 234, 234
4280 DATA  234, 234, 32, 159, 255,
 165, 197, 201, 62, 208
4290 DATA  3, 76, 252, 205, 201, 5
6, 208, 5, 169, 1
4300 DATA  76, 243, 204, 201, 27,
240, 6, 32, 89, 204
4310 DATA  76, 210, 204, 169, 8, 1
41, 21, 207, 169, 1
4320 DATA  168, 174, 21, 207, 32,
186, 255, 173, 22, 207
4330 DATA  162, 32, 160, 207, 32,
189, 255, 173, 20, 207
4340 DATA  208, 6, 32, 213, 255, 7
6, 32, 205, 32, 3
4350 DATA  204, 169, 250, 166, 252
```

Program Listing (cont.)

```
, 164, 253, 32, 216, 255
4360 DATA  104, 104, 76, 252, 205,
 10, 24, 105, 48, 170
4370 DATA  105, 2, 96, 169, 0, 141
, 20, 207, 76, 183
4380 DATA  204, 169, 1, 208, 246,
173, 8, 207, 72, 173
4390 DATA  9, 207, 72, 32, 197, 20
0, 32, 158, 195, 104
4400 DATA  133, 253, 104, 133, 252
, 160, 0, 177, 252, 145
4410 DATA  250, 200, 192, 8, 208,
247, 76, 14, 201, 173
4420 DATA  141, 2, 41, 4, 208, 3,
76, 39, 206, 76
4430 DATA  22, 203, 173, 21, 208,
9, 2, 76, 17, 206
4440 DATA  76, 197, 200, 32, 90, 2
02, 169, 0, 133, 198
4450 DATA  76, 230, 193, 32, 230,
193, 169, 0, 133, 198
4460 DATA  104, 104, 76, 114, 202,
 169, 0, 141, 16, 208
4470 DATA  141, 21, 208, 141, 23,
208, 141, 29, 208, 32
4480 DATA  252, 197, 169, 13, 141,
 248, 7, 141, 249, 7
4490 DATA  169, 1, 141, 39, 208, 1
41, 40, 208, 169, 40
4500 DATA  141, 0, 208, 169, 66, 1
41, 1, 208, 169, 56
4510 DATA  141, 2, 208, 169, 146,
141, 3, 208, 234, 96
4520 DATA  234, 234, 234, 234, 234
, 234, 234, 234, 32, 186
4530 DATA  202, 32, 137, 205, 32,
33, 202, 162, 8, 189
4540 DATA  255, 47, 157, 239, 207,
 202, 208, 247, 32, 124
4550 DATA  192, 32, 76, 193, 32, 1
49, 199, 173, 21, 208
4560 DATA  41, 1, 141, 21, 208, 32
, 232, 201, 32, 88
4570 DATA  194, 32, 2, 201, 32, 12
7, 198, 76, 229, 205
4580 DATA  169, 2, 133, 2, 160, 0,
 162, 0, 202, 208
4590 DATA  253, 136, 208, 248, 198
, 2, 208, 242, 76, 192
4600 DATA  205, 41, 2, 141, 21, 20
8, 76, 197, 200, 169
4610 DATA  0, 133, 170, 173, 141,
2, 41, 4, 133, 170
4620 DATA  76, 50, 206, 173, 21, 2
08, 41, 2, 141, 21
4630 DATA  208, 76, 57, 205, 169,
20, 141, 24, 208, 76
4640 DATA  53, 205, 169, 20, 141,
24, 208, 76, 45, 205
4650 DATA  3, 32, 8, 32, 1, 32, 18
, 32, 1, 32
4660 DATA  3, 32, 20, 32, 5, 32, 1
8, 32, 32, 7
4670 DATA  32, 5, 32, 14, 32, 5, 3
2, 18, 32, 1
4680 DATA  32, 20, 32, 15, 32, 18,
 32, 39, 54, 52
4690 DATA  16, 32, 18, 32, 5, 32,
19, 32, 19, 32
4700 DATA  32, 1, 32, 14, 32, 25,
32, 32, 11, 32
4710 DATA  5, 32, 25, 32, 32, 32,
20, 32, 15, 32
4720 DATA  32, 19, 32, 20, 32, 1,
32, 18, 32, 20
4730 DATA  169, 48, 141, 17, 207,
234, 234, 234, 169, 147
4740 DATA  32, 210, 255, 169, 142,
 32, 210, 255, 169, 8
4750 DATA  32, 210, 255, 169, 0, 1
41, 32, 208, 141, 33
4760 DATA  208, 169, 64, 133, 197,
 162, 40, 189, 105, 206
4770 DATA  157, 183, 5, 169, 1, 15
7, 183, 217, 189, 65
4780 DATA  206, 157, 255, 3, 169,
7, 157, 255, 215, 202
4790 DATA  208, 231, 165, 197, 201
, 64, 240, 3, 76, 236
4800 DATA  206, 32, 89, 204, 162,
40, 189, 255, 3, 73
4810 DATA  128, 157, 255, 3, 202,
208, 245, 76, 208, 206
4820 DATA  173, 111, 201, 72, 208,
 3, 32, 254, 193, 104
4830 DATA  9, 1, 141, 111, 201, 23
4, 234, 76, 192, 205
4840 DATA  169, 1, 166, 2, 160, 1,
 32, 186, 255, 169
4850 DATA  7, 162, 35, 160, 207, 3
2, 189, 255, 169, 0
4860 DATA  133, 250, 169, 192, 133
, 251, 169, 250, 162, 0
4870 DATA  160, 207, 76, 216, 255,
 71, 69, 78, 39, 54
4880 DATA  52, 46, -1
```

PROGRAMMED TO WIN!



Win a unique Austin Metro worth £5,500 with the October Issue of

64 TAPE COMPUTING

Get motoring this Winter with a special edition Austin Metro. You choose the trim and fittings for a real eye-catcher. All you have to do is become our Tape Magazine Computer Champion.

We are having a two-part competition in our October and November issues.

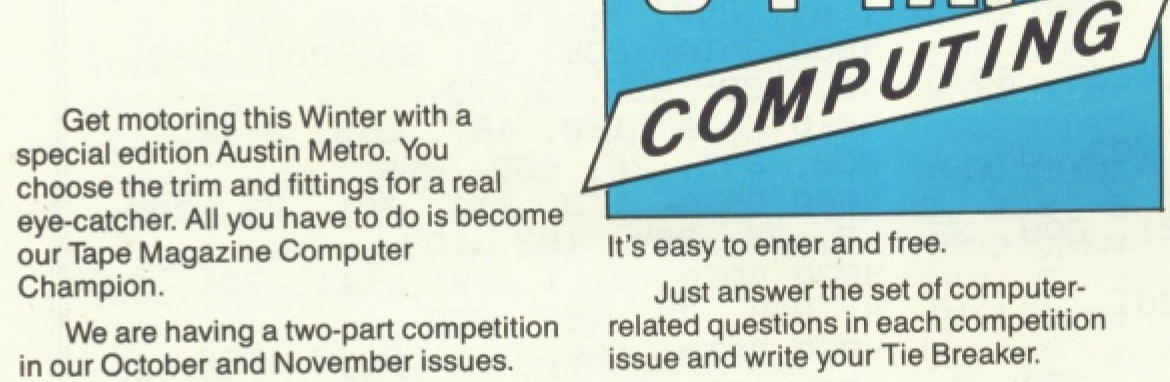
It's easy to enter and free.

Just answer the set of computer-related questions in each competition issue and write your Tie Breaker.

Can you make the grade and become our Computer Champion?

The competition closes on **30 November 1985.**

Get motoring and order your issues now ... before they sell out.



ROULETTE TREASURE ISLAND STRATEGY SPECIAL

Argus Press Software Group

Available every month from all branches of W H Smiths, John Menzies and other leading newsagents.

Argus Tape Magazines produced by
64 Tape Computing
Infonet Limited
Times House 179 The Marlowes
Hemel Hempstead Herts HP1 1BB

GREAT GAMES : GREAT UTILITIES : GREAT VALUE

COMMODORE PLUS 4 AND C16

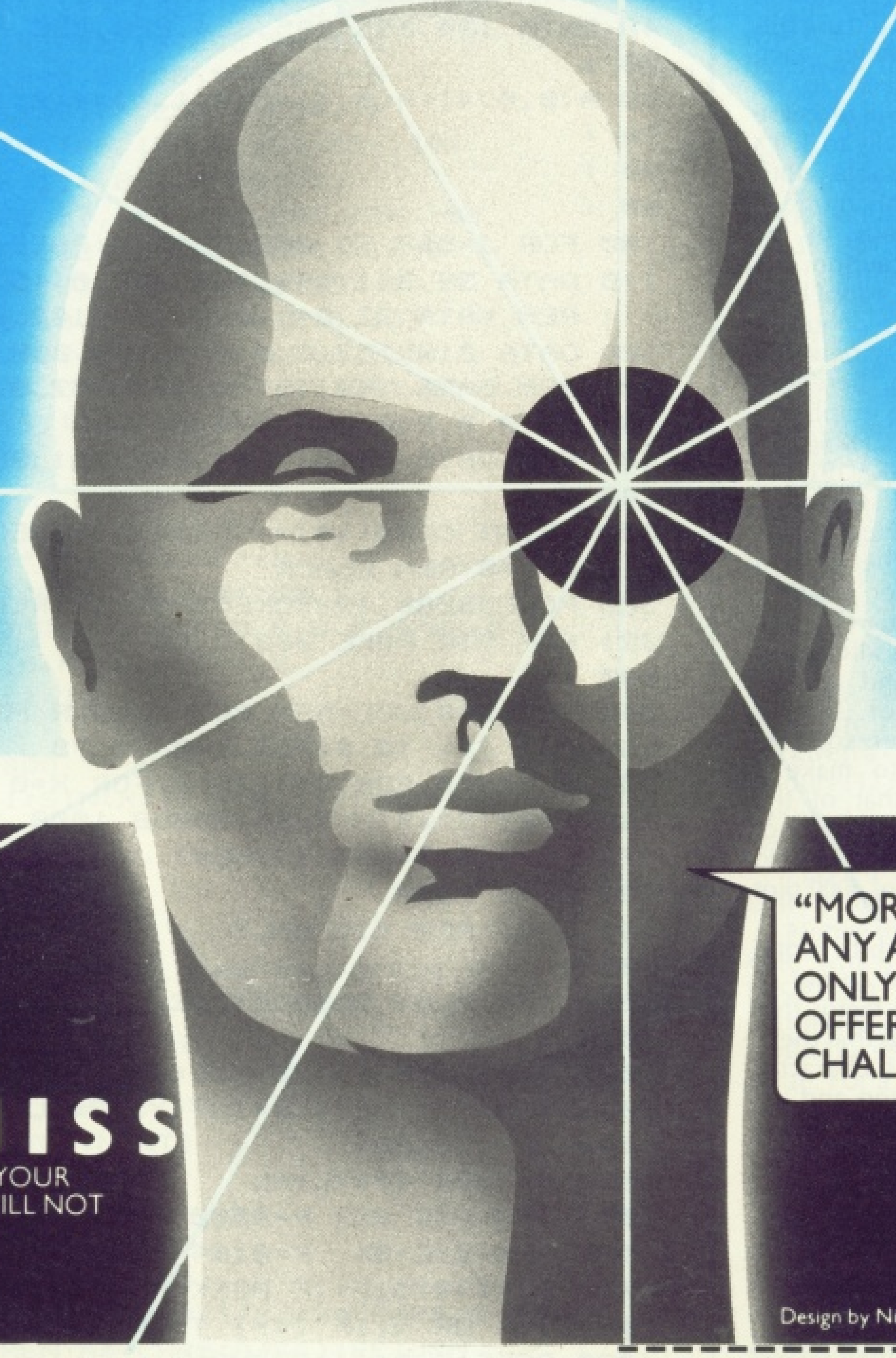
CAN YOU MEET THE CHALLENGE OF:–

HIT & MISS

HIT & MISS IS THE CHALLENGE
BUT – THE QUESTION IS –
CAN YOU MEET IT?

FREE A2 POSTER
FREE PACKAGE & POSTAGE
NEW – IDEAS IN ACTION
ONLY – FOR PLUS 4/C16

TAPE
£7.99



"MORE ADDICTIVE THAN ANY ARCADE GAME, THE ONLY C16/PLUS 4 SOFTWARE OFFERING A REAL CHALLENGE"

WARNING:– PLAYING

HIT & MISS

COULD SERIOUSLY DAMAGE YOUR HEALTH. BUT IT CERTAINLY WILL NOT DAMAGE YOUR WEALTH

Design by Nigel Widdowson Graphic Design. Leeds

HIT & MISS DEFINITELY IS A MIND CHALLENGE OF MEGA PROPORTION THAT ALL YOUR FAMILY AND FRIENDS CAN TAKE PART IN. FROM SINGLE PLAY TO DOUBLES IT WILL STRETCH YOUR MENTAL DEXTERITY TO THE FULL.......

BUT REMEMBER:–
"IT TAKES MORE THAN JUST GUESSWORK TO BECOME A GENIUS WITH **HIT & MISS**"

VENTUREGATE LTD

AVAILABLE AT MOST GOOD SOFTWARE OUTLETS

ORDER YOUR COPY:-
FROM:- VENTUREGATE LTD., 17 HAROLD ROAD, W. YORKSHIRE, LEEDS LS6 1PR.

NAME (BLOCK LETTERS) ______

ADDRESS ______

COUNTY ______ POSTCODE ______

PLEASE SEND ME ______ COPY/IES OF **"HIT & MISS"** AT £7.99 COMPLETE WITH A FREE POSTER WITH EVERY COPY.

(FREE A2 POSTER AND FREE P & P)

RELIABLE ROUTINES

Mike Hart provides a couple of handy plotting routines for the Vic 20 and C64.

AT ONE TIME OR ANOTHER, MANY people must have experimented with their machines to see if there is an easy way to draw graphs or plots on the screen. There are three ways in which this can be done and I will call them low resolution, medium resolution and high resolution respectively.

In low-resolution plotting, one merely takes the screen as a grid and plot points using asterisks or a similar graphic characters. In the case of a C64 this would obviously be a grid of 40 × 25 giving 1000 potential plot points.

If one wished to use a high resolution screen then on the C64 one would use bit-mapped graphics which allows each individual 'dot' on the screen to be controlled. Using this mode will increase the resolution quite dramatically to 320 × 200 which is 64000 addressable dots. However using the C64 in this mode is quite complicated and needs to be approached with a degree of caution. The approach which I am going to adopt here is one that gives pleasant 'chunky' graphics points and is half-way between these two extremes and which, therefore, I shall call medium-resolution graphics.

If you were to PRINT CHR$(161), CHR$(188), CHR$(172) and CHR$(191) as a brief experiment you would see that a graphics character is generated which consists of a quarter square either by itself or in combination with another 'quarter-square' to make up a half square. There are 14 of these combinations altogether and together with space and reverse space we have a total of sixteen permutations of 'quarter-square' graphics. This enables us to make a resolution which is twice that of the normal screen and therefore instead of having 40 × 25 we can increase this to 80 × 50 giving us a resolution of 4000 plot points.

Let us suppose that we wish to plot a point at the bottom left hand corner of the screen. The complicating factor that we have to take into account is that there might already be a graphics character already occupying that position and we would wish to preserve the point that it represents. The trick here is to PEEK the screen at that particular point, read the value of the character at that point, look up its value in a table of potential values and then work out from the same table the value of the new character to be poked back onto the screen which both plots the 'new' point and the value of the existing point. If this is a little hard to visualise then any of Raeto West's books have an excellent explanation under the

Program Listing

MEDIUM-RES PLOT

```
1 REM   *** MEDIUM-RES PLOT (BASIC) ***
2 :
3 REM    **        M. C. HART          **
4 :
10 DIM C(15),A(1,1)
20 FORJ=0 TO 15:READ C(J):NEXT
30 DATA 32,123,108,98,126,97,127,252
40 DATA 124,255,225,254,226,236,251,160
50 :
60 A(0,0)=1:A(0,1)=2:A(1,0)=4:A(1,1)=8
70 :
80 :
90 :
100 FOR J=948 TO 969:READ X:POKE J,X:NEXT
110 DATA 32,241,183,138,162,0,157,0
111 REM DATA 32,241,215,138,162,0,157,0
120 DATA 216,157,0,217,157,0,218,157
121 REM DATA 148,157,0,149,157,0,150,157
130 DATA 0,219,202,208,241,96
131 REM DATA 0,151,202,208,241,96
140 :
300 PRINT CHR$(147);CHR$(144):REM BLACK
310 COL=0:REM BLACK
320 POKE53280,12:POKE53281,12:REM GREY2
321 REM VIC POKE 36879,25
330 :
340 SYS 948,COL:REM SET COLOUR MEMORY
350 FOR J=1 TO 2:FOR X=0 TO 79
351 REM VIC FOR J=1 TO 2:FOR X=0 TO 39
360 Y=22+21*SIN(X/8):GOSUB 1000
370 NEXT X:T=NOT(T):NEXT J
380 GOTO 2000
381 END: REM VIC ONLY
995 :
1000 REM PLOT SUBROUTINE
1010 :
1020 XL=INT(X/2):XS=X-2*XL
1030 YL=INT(Y/2):YS=INT(Y)-2*YL
1040 N=A(YS,XS)
1050 P=1984-40*YL+XL
1051 REM VIC 8K+ P=4580-22*YL+XL
1052 REM VIC<8K  P=8164-22*YL+XL
1060 FORI=0TO15:IF PEEK(P)<>C(I)THEN NEXT
1070 I=I OR N:IF T=-1 THEN I=I AND NOT(N)
1080 POKE P,C(I):RETURN
1090 :
2000 :
```



Program Listing

```
2001 REM *** MEDIUM-RES PLOT (MC)   ***
2002 REM
2003 REM  **  BY M. C. HART  **
2004 REM
2010 FOR J=828 TO 990:READ X:T=T+X
2020 POKEJ,X:NEXT:IF T=22553 THEN 3000
2030 PRINT "DATA ERROR!":END
2031 DATA 169,0,133,254,169,1,133,253
2032 DATA 165,251,201,80,176,56,165,252
2033 DATA 201,50,176,50,169,50,229,252
2034 DATA 70,251,38,254,106,38,254,133
2035 DATA 252,10,10,101,252,10,10,38
2036 DATA 253,10,38,253,234,234,234,133
2037 DATA 252,166,254,189,159,3,133,254
2038 DATA 164,251,177,252,162,15,221,163
2039 DATA 3,240,4,202,16,248,96,173
2040 DATA 158,3,240,6,138,5,254,170
2041 DATA 208,8,138,73,255,5,254,73
2042 DATA 255,170,189,163,3,164,251,145
2043 DATA 252,96,1,1,2,4,8,32
2044 DATA 126,123,97,124,226,255,236,108
2045 DATA 127,98,252,225,251,254,160,0
2046 DATA 32,241,183,138,162,0,157,0
2047 DATA 216,157,0,217,157,0,218,157
2048 DATA 0,219,202,208,241,96,32,235
2049 DATA 183,134,252,165,20,133,251,32
2050 DATA 241,183,142,158,3,32,180,3
2051 DATA 76,60,3
2060 :
3000 PLOT=970
3010 PRINT:INPUT"BACKGROUND";B
3020 PRINT:INPUT"PEN COLOUR";COL
3030 PRINT:INPUT"TWO NUMBERS E.G 2,3";X,Y
3040 POKE 53280,B:POKE 53281,B
3050 PRINT CHR$(147)
3060 :
4000 FORJ=1 TO 800
4010 SYS(PLOT)(1+SIN(X*J))*40,(1+COS(Y*J))*25,1,COL
4020 NEXT
4030 :
63000 REM SCREEN-DUMP
63010 :
63020 OPEN6,4,6:PRINT#6,CHR$(22):OPEN4,4
63030 FORI=0 TO 24:FOR J=0 TO 39
63040 A=PEEK(1024+I*40+J)
63050 GOSUB63100
63060 PRINT#4,A$;B$;C$;
63070 NEXTJ:PRINT#4:NEXTI
63080 PRINT#4:CLOSE4:END
63090 :
63100 A$="":B$="":C$=""
63110 IFA<=128 THEN 63130
63120 A$=CHR$(18):C$=CHR$(146):B$=CHR$(A-64):RETURN
63130 IF A<32 THEN A$=CHR$(A+64):RETURN
63140 IF A>31 AND A<64 THEN B$=CHR$(A):RETURN
63150 IFA >63 AND A<96 THEN B$=CHR$(A+128):RETURN
63160 B$=CHR$(A+64):RETURN
```

heading 'double density graphics' complete with diagrams and very full explanations of the manipulations involved.

Lines 1-1090 represent an implementation of medium-resolution graphics using BASIC only. This is fairly slow but is speeded up somewhat by making use of a machine code routine to poke the colour RAM so that we are not concerned with two POKEs for every point plotted. The program is given by default for the C64 but the changes needed for the VIC's are REMmed immediately after each of the affected lines.

Lines 10-60 are concerned with constructing a small 'look up" table.

Lines 100-140 read in the machine code for subsequent changes of colour RAM.

The major routine is in lines 300-380 which computes a sine curve and then plots it (on the first of the J loops) and then 'unplots' it (on the second of the J loops).

The procedure for VIC owners is to delete the following lines: 110,120,130, 320,350,380,1050. Now take the lines that immediately follow these i.e. 111,121,131, 321,351,381,1051-1052 and remove the REM portion of the statement that made these lines inoperative in the C64 version. You will also have to decide which version of VIC expansion you are operating with, as the screen shifts from $1E00 to $1000 just to make life confusing. Select either line 1051 or 1052.

Line 380 having been deleted makes the VIC version end in line 381.

Owners of the C64 now have a machine code version of the above which plots a Lissajous figure in the shape of a butterfly. An illustration of this is also given.

Lines 2000-2051 read the machine code into the cassette buffer. Then you are given a choice of background and pen colours (12 for background and 0 for pen colour i.e., black on light grey is my personal favourite).

PLOT is defined as 970 and this is the entry point.

Notice the complicated formula in line 4010 which actually follows the rubric SYS(PLOT) x,y,on/off, colour. The first computed value is obviously the x value whilst the second computed value is the y value. For a 'plot' we would then have a 1 and for an 'unplot' or 'erase' we would have a 0. Finally, we have the colour intended for the plot. You can obviously experiment with it as much as you like.

The origin is in the bottom left-hand corner (SYS(PLOT)0,0,1,0 whilst the opposite corner is SYS (PLOT)79,49,1,0).

Finally, lines 63000-63160 constitute a screen dump, written in BASIC but more than adequate.

Nick Hampshire reveals the mysteries of the TED chip in the C-16 and Plus/4.

TED CHIP

THE GRAPHICS DISPLAY, SOUND GENeration and internal clock/timers of the C-16 and Plus/4 computers are controlled by a single integrated circuit, the so-called TED chip. This is a complex device, and, unfortunately, rather difficult to use.

An equally unfortunate circumstance is that no information on the use of this chip is provided in Commodore's manual. This is presumably in the belief that the graphics and sound commands supplied in the extended Basic are adequate. However, most advanced programmers, especially those writing machine code programs, will want direct access to the registers of this device.

The TED chip is a rather strange device. It is located in the middle of the kernal ROM area and overlays this ROM so that the ROM area covered by TED is inaccessible. In addition, the TED registers are not grouped in one continuous area of memory. We located TED registers in the area $FD00 to $FF3F. The reason for this is obscure and probably related to a quirk in the chip's design.

In operation the TED chip is not unlike the VIC and SID chips in the C64 and it is worth studying one of the advanced books on the 64 (for instance *Advanced Commodore 64 Graphics and Sound* and *The Commodore 64 Kernal and Hardware Revealed* – both by Nick Hampshire).

The following table shows the locations in TED which we have uncovered together with the function of each register and the bits within each location.

TED Graphics/sound/keyboard control.

FF00		— Timer 1 low
FF01		— Timer 1 high
FF02		— Timer 2 low
FF03		— Timer 2 high
FF04		— Timer 3 low
FF05		— Timer 3 high
FF06		— Video control 1
	Bit 7	Not used
	6	I = extended background
	5	1 = Bit map
	4	0 = blank screen
	3	1 = 25 lines, 0=24 lines
	2-0	Vertical smooth scroll pos
FF07		— Video control 2
	Bit 7-5	Not used
	4	1 = Multi-colour
	3	1 = 40 columns, 0 = 38 columns
	2-0	Horizontal smooth scroll pos
FF08		— Out : keyboard column or joystick (FD or FA) In:keyboard row or joystick switches
FF09		— Interrupt control
	Bit 5	T3 has run out
	4	T2 has run out
	3	T1 has run out
	1	Raster compare occurred
FF0A		— Interrupt enable
	Bit 5	1 = T3 enable
	4	1 = T2 enable
	3	1 = T1 enable
	1	1 = Raster enable
FF0B		— Raster compare low byte
FFOC		— Screen offset from base for cursor (high byte
FF0D		— Screen offset from base for cursor (low byte)
FF0E		— Voice 1 low byte
FF0F		— Voice 2 low byte
FF10		— Bits 1-0:Voice 2 high
FF11		— Sound control
	Bit 7	Disable sound
	6	1 = Noise on voice 2
	5	1 = Voice 2 tone enable
	4	1 = Voice 1 enable
	3-0	Volume (0-8 only)
FF12		
	Bit 3	Bit map base
	2	1 = chars from ROM,0=chars from RAM
	1-0	Voice 1 high
FF13		— Character base address
	Bit 7-3	Address of UDSc (2K steps)
	2	1 =lower case
FF14		— Screen base address
	Bit 7-3	Address of colour memory (2K steps, screen 1K above colour)
FF15		— Background colour
	Bit 7	Not used
	6-4	Luminance (0-7)
	3-0	Colour (0-15)
FF16		— Extended back 1/M col 1
	Bit 7	Not used
	6-4	Luminance (0-7)
	3-0	Colour (0-15)
FF17		— Extended back 2/M col 2
	Bit 7	Not used
	6-4	Luminance (0-7)
	3-0	Colour (0-15)
FF18		— Extended back 3
	Bit 7	Not used
	6-4	Luminance (0-7)
	3-0	Colour (0-15)
FF19		— Border colour
	Bit 7	Not used
	6-4	Luminance (0-7)
	3-0	Colour (0-15)
FF1C		— Bit 0: Raster position high bit
FF1D		— Raster position low byte
FF3E		— ROM in when written to
FF3F		— ROM out when written to
FD10		— Bit 2:cassette switch sense (0=down)
FF30-FD3F		— =no key, FF=key down

Scratchpad

A fistful of DATAs — more of your useful programming quickies presented by Max Phillips.

WELCOME TO THE PAGE FOR YOUR programming bits and pieces. We're interested in anything, useful or amusing, from a few POKEs to a short utility and we'll pay for anything we use. Just send us your program (either as a listing or on cassette or disk) and some notes as to what it does and how it does it. Post your contributions to Scratchpad, Your Commodore, No 1 Golden Square, London W1R 3AB.

That '!'?! Bleepin Keyboard

Andreas Weinand kicks off this month with a machine code utility that produces a beep every time a key is pressed. It might be a help for anyone who has a dicky keyboard or can't manage two finger typing but watch out – it does mean everyone else can hear what your typing speed is like!

Andreas' program takes over the C64's regular interrupt and uses voice 1 on the SID chip to provide an expensive sounding beeper.

Clean Living

Wash is a handy disk utility supplied by Clifford Hanger of Manningtree in Essex. It lets you step through all the files on a disk one by one using the space bar. You can delete the current file by pressing S or rename it with R. Even better, pressing V lets you take a peek at what's in the file so you can figure out what it is and then scratch it if you don't need it.

The VIEW command works with all files though it cuts out any control characters and replaces them with full stops. Even so, you should be able to get a good idea of what the file is. One problem – after viewing a file, Wash sometimes goes back to the title of the disk – just hit space a few times to go round again. It just goes to show – you can write a useful utility in Basic!

Oh No, Not Again...

Printing at a particular position on the screen is still the most talked about subject in C64 programming. Graham Blighe of Eastleigh in Hampshire has pointed out that David Read's routine in Your 64 issue 13 won't let you print on the top line of the screen because you've got to POKE 214,Y-1 and then PRINT. Oh yes – I wondered when someone would spot that!

Graham has a neat solution – call the routine in the kernal ROM which recalculates the cursor position. The PRINT-AT routine then becomes:

POKE 211,X:POKE 214,Y:SYS 58732

Meanwhile, Asmat Ullah from Glasgow has a fiendishly clever machine code solution. It's only 12 bytes of code so it should be no trouble to include it in your initialisation section. Once loaded, you can print anywhere on the screen with:

SYS(828)X,Y;"MESSAGE"

Neat huh? Asmat does point out that there's no error checking in the routine in order to keep it short so you should make sure that X is less than or equal to 40 and Y is less than or equal to 25. Even so, it has to be the best yet. Unless, of course, you know different...

```
1 REM PRINT-AT ASMAT ULLAH
2 FOR X=828 TO 839:READ Y:POKE
X,Y:NEXT
3 DATA 32,234,183,164,20,24,32,240,255.7
```

Clean Cut Characters

Paul Barnham of Darwen in Lancashire has sent in a delightful utility that replaces all those hard to read control codes in listings with clear mnemonics such as [CLR9], [RED], [UP] and so on. Just enter POKE 56,15:CLR and run the listing below. You'll find LIST a changed command...

ANDREAS WEINAND LISTiNG

```
100 REM KEY BEEP BY ANDREAS WEINAND
110 REM START WITH SYS 832
120 FOR I=832 TO 940
130 READ X: POKE I,X: S=S+X: NEXT
140 DATA169, 15,141, 24,212,169, 64,141,
 5,212,169, 68
150 DATA141,  6,212,169, 69,162,157,141,
  1,212,142,  0
160 DATA212,169,100,162,  3,141, 20,  3,
142, 21,  3, 96
170 DATA165,203,201, 64,208,  3, 76, 49,
234,162, 17,142
180 DATA  4,212,141, 87, 27,169,135,162,
  3,141, 20,  3
190 DATA142, 21,  3,169, 12,141, 86, 27,
 76, 49,234,165
200 DATA203,205, 87, 27,208, 12,174, 86,
 27,202,240,  6
210 DATA142, 86, 27, 76, 49,234,169,  0,
141,  4,212,169
220 DATA100,162,  3,141, 20,  3,142, 21,
  3, 76, 49,234
230 DATA 0
240 IF S<>11366 THEN PRINT "ERROR IN DAT
A":END
250 PRINT "OK"
```

CLIFFORD HANGER LISTING

```
10 REM DIRECTORY WASH: C HANGER 1985
20 PRINT "[CLR]";TAB(18);"[RVS]WASH[OFF]":PRINT
30 PRINT "... FOR CLEANER DIRECTORIES!"
40 REM READ DIR
50 OPEN2,8,15
60 OPEN 1,8,0,"$0"
70 GET#1,A$,B$
80 GET#1,A$,B$,A$,B$
90 C=0:IF A$<>"" THEN C=ASC(A$)
100 IF B$<>"" THEN C=C+ASC(B$)*256
110 C$=STR$(C):C$=LEFT$("    ",4-LEN(C$)
)+C$
120 GET#1,B$:IF ST<>0 THEN PRINT:PRINT C
$;" BLOCKS FREE":CLOSE 1:GOTO60
130 IF B$<>CHR$(34) THEN 120
140 F$=""
150 GET#1,A$:IF A$<> CHR$(34) THEN F$=F$
+A$:GOTO 150
160 GET#1,A$:IF A$=" " THEN 160
170 T$=A$
180 GET#1,A$: IF A$<>"" THEN T$=T$+A$:GO
TO180
190 IF C=0 THEN PRINT:PRINT "[RVS]TITLE[OFF] ";F
$:PRINT:GOTO80
200 PRINT F$;TAB(18);"[RVS]BLKS[OFF]";C$;TAB(27)
;"[RVS]TYP[OFF] ";LEFT$(T$,3)
210 PRINT:PRINT "[RVS]S[OFF]CRATCH [RVS]R[OFF]ENAME [RVS]V[OFF]I
```

```
EW ▒Q▒UIT OR ▒SPACE▒ ";
220 REM GET COMMAND
230 GET K$: IF K$="" THEN 230
240 PRINT K$:PRINT
250 IF K$="Q" OR K$="●" THEN CLOSE 1:PRI
NT"□":CLOSE 2:END
260 IF K$=" "THEN 300
270 IF K$="S" OR K$="♥" THEN PRINT#2,"S:
"+F$;
280 IF K$="R" OR K$="_" THEN GOSUB330
290 IF K$="V" OR K$="×" THEN GOSUB360:GO
TO200
300 IF ST=0 THEN80
310 CLOSE 1:CLOSE 2:END
320 REM RENAME FILE
330 INPUT"NEW NAME ";A$
340 IF A$="" THEN PRINT:RETURN
350 PRINT#2,"R:"+A$+"="+F$:PRINT:RETURN
360 REM VIEW FILE
370 PRINT:PRINT"VIEWING ";F$
380 PRINT:PRINT"▒SPACE▒ TO PAUSE, ▒A▒ TO
 ABORT":PRINT
390 OPEN 5,8,5,"0:"+F$+","+LEFT$(T$,1)+"
,R"
400 GET#5,A$: IF ST=0 THEN 430
410 CLOSE 5:POKE 144,0
420 PRINT:PRINT"▒FILE ENDS▒":PRINT:RETUR
N
430 IF A$<>"" THEN A$=CHR$(ASC(A$)AND 12
7):IF A$<CHR$(32) THEN A$="."
440 PRINTA$;
450 GETK$:IF PS=1 AND K$="" THEN 450
460 IF K$=" " THEN PS=ABS(1-PS)
470 IF K$<>"A" AND K$<> "♠" THEN 400
480 CLOSE 5:POKE 144,0
490 PRINT:PRINT"▒ABORTED▒":PRINT:PRINT:R
ETURN
```

PAUL BARHAM LISTING

```
10 FORX=0TO217:READA:T=T+A:POKE40704+X,A
:NEXT
15 IFT=25486THENSYS40704:END
17 PRINT"ERROR":STOP
20 DATA169,11,141,6,3,169,159,141
30 DATA7,3,96,133,251,72,138,72,152
40 DATA72,164,15,192,1,240,23,162,0
50 DATA189,89,159,197,251,240,24,232,232
60 DATA232,232,189,89,159,201,0,240,3
70 DATA76,29,159,104,168,104,170,104,165
80 DATA251,76,26,167,169,91,32,71,171
90 DATA232,189,89,159,48,3,76,59,159
100 DATA41,127,32,71,171,169,93,32,71
110 DATA171,104,168,104,170,104,76,0,167
120 DATA19,72,79,205,147,67,76,210,144
130 DATA66,76,203,5,87,72,212,28,82
140 DATA69,196,159,67,89,206,156,80,85
150 DATA210,30,71,82,206,31,66,76,213
160 DATA158,89,69,204,145,85,208,88,17
170 DATA68,206,88,29,82,71,212,157,76
180 DATA71,212,133,70,177,88,134,70,179
190 DATA88,135,70,181,88,136,70,183,88
200 DATA137,70,178,88,138,70,180,88,139
210 DATA70,182,88,140,70,184,88,18,82
220 DATA79,206,146,82,79,198,129,79,82
230 DATA193,149,66,82,206,150,76,82,196
240 DATA151,71,82,177,152,71,82,178,153
250 DATA76,71,210,154,76,66,204,155,71
260 DATA82,179,0
```


YOUR COMMODORE — CLASSIFIED ADVERTISEMENT — ORDER FORM

Advertise nationally in these columns for only 40p per word (min charge 15 words). Simply print your message in the coupon and send with your cheque or postal order made payable to **Argus Specialist Publications Ltd** to:

CLASSIFIED DEPARTMENT
YOUR COMMODORE
No 1 Golden Square, London W1
Tel: 01-437 0699

Name ....

Address ....

....

....

Tel no (day) ....

ADVERTISERS PLEASE ADD 15% VAT

Please place my advert in **YOUR COMMODORE** for .... issues.

Please indicate no. of insertions required.

YOUR COMMODORE

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699
EXT. 322

Send your requirements to:
DAVID HORNE
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

HARDWARE

NEW STAR SG-10C PRINTER

ONLY £249 TO INCLUDE VAT & NEXT DAY DELIVERY BY SECURICOR

- ★ 100% compatible with your Commodore computer.
- ★ 120cps draft mode.
- ★ Tractor and friction feed standard.
- ★ Near Letter Quality mode included.
- ★ 192 graphic mode characters.
- ★ 192 business mode characters.
- ★ 90 italic characters.
- ★ 90 NLQ characters.
- ★ Normal 80 cpl. Condensed 136 cpl.
- ★ Enlarged 2X, 3X & 4X.
- ★ Supplied with interface/cable and mains plug READY TO GO.

STRONG COMPUTER SYSTEMS
BRYNA COTTAGE, PENIEL, CARMARTHEN, DYFED SA32 7DJ
TEL: (0267) 231246 FOR ASSISTANCE!!! C64

★ BACK-UP-SYSTEMS FOR CBM-64 & VIC-20 ★

The BACK-UP-BOARD Mk. 1 is the fastest, and easiest way to back up your tape programs. With this board you can connect a second CBM cassette deck to your computer. You can copy a program on the second deck while it loads from the first. You can back up all fast loaders, data files, etc. BACK-UP-BOARD Mk1 £10 inc. P&P.

BACK-UP-BOARD Mk.2

This new version of our famous BACK-UP-BOARD now works two ways. Switched to 'NORMAL' it operates as the Mk.1 Switched to 'GENERATE', '64' owners can use our special transfer software to regenerate a copy of a program on the second deck as it loads from the first. This method can produce a better quality copy than the original. BACK-UP-BOARD Mk.2 and software £14.50 inc. P&P.
Conversion service. Return your Back-up-Board, if purchased form us, to be modified to Mk.2 for £4.50 including software and P&P.

AZIMUTH HEAD ALIGNMENT TAPE BY INTERCEPTOR FOR '64

Check out your cassette decks with this excellent program. £8.50 inc. P&P.

ROM SWITCH/RESET BOARD

Fit this board in the expansion port of your '64' and you can reset ALL programs. Resets programs that a standard reset switch can't. A socket for a 2764 8K eprom is also fitted. £10 inc P&P.

JOYSTICK PORT SWITCH FOR CBM 64

Use this adaptor to allow you to change from port 1 to port 2 at the push of a button. No more frustrated fumbling. Play two player games with one joystick, just push button for player two's turn. Includes built-in rapid fire switch for those impossible shoot 'em up games. For the '64' owner who thought they had everything. £16.50 inc P&P.

Send your cheque/PO to:
TURBOTRONIC (Dept. YC)
46 Ripon Street, Parkinson Lane, Halifax, West Yorkshire HX1 3UG
Tel: (0422) 52020 (Europe add 50p outside Europe add £1.50)

Super Breaker Back-Up Board

For the (Commodore 64, CBM 128 and VIC-20) When a second CBM type cassette unit is connected to the board it will make security back-up copies of cassette based software, including Turbo/Fast Loaders, Basic and/or Machine Code and Data Files. No additional software is needed. Very User Friendly, the back-up copy is made on the second cassette while the original is loading on the first. Price at £8.95 Plus P&P.

RESET CARTRIDGE for CBM 64. This cartridge will reset all, yes all, memory resident programs. Including all programs where standard reset switches foul. Price £7.95 + P&P.
P&P 55p UK (Europe add £1.00, elsewhere add £2.00). Send cheque/postal order, or a stamp for details, to:

C. Kent, 26 Northcape Walk, Corby, Northants.
For enquiries telephone (0536) 742622

COMMODORE C16

DUAL CASSETTE INTERFACE

Transfer any C16 cassette program, turbos, m/code etc. to a second datasette.
Interface I for C16 and +4: **£11.95** inc P&P. Interface II for C16, +4, CBM 64 and Vic 20: **£14.95** inc P&P.
Joystick Interface, connects any joystick to C16: **£3.95** inc P&P.
Cassette Interface, connects any C2N cassette to C16: **£3.25** inc P&P.
Cheques/POs to:
J. FITZPATRICK (YC),
1 Alveley Ave, Withington, Manchester M20 9UD.

REPAIRS

COMMODORE REPAIRS

Commodore repairs by Commodore approved engineers. Repair prices: Vic 20 modulator **£8.05**, Vic 20 **£15.00**, CBM 64 from **£20.00**, C2N from **£9.00**, 1541 from **£20.00**, C64 psu **£13.50**, Spectrum **£16.50**, Dragon **£32.00**, Electrons, Atari.
For more information SAE or telephone:
G.C. BUNCE & SON,
36 Burlington Road, Burnham, Bucks SL1 7BQ.
Tel: (06286) 61696.

For fast reliable repair service send you rVIC-20 modulator (7.50), and Vic-20 from £12.50 and Commodore 64 from £16.00 plus postage and packing, to:

HEMEL COMPUTER CENTRE
52 High Street
Hemel Hempstead
Herts HP1 3AF
Tel: (0442) 212436

TO FILL THIS SPACE RING 01-437 0699

FOR SALE

ELECTRICAL — CONTRACTORS!!
Comply with the 15th edition with my cable-sizing program for the CBM-64. Calculates cable capacity, CPC size, maximum allowable runs for volt-drop and shock protection etc. Readout indicates if cable doesn't comply. **£6.95.** Cheques/PO with order to:
TONY SPRY,
13 Tudor Drive,
Barnstaple, Devon.

CBM 64 quality colour monitor style T.V. complete system* including good software library for sale. £350.00 O.N.O. Ring Bentley (0420) 23565 (Hants). *Tape based.

1526 PRINTER. Immaculate condition. Hardly Used. Boxed bargain at £220 ONO. Telephone 0685 812219 After 7pm.

MOONRAKER £6.95
CBM 64 Space Game features fast
- Fly-Into Rolling Radar Display
- Infra Red Target Screen

Altitude, Time, Score, Indicators, Fuel/Purchase Option, Promotion.
Send your name, address & PO/cheque to:
J & J Gilmour Ltd., Dept YC1085
4 Copse Close, Liss, Hampshire GU33 7EN, England GU33 7EN.
Tel: 0730-893430

SOCCER MANAGER (Commodore 64). Manage your own team!! Four divisions. F.A. Cup. Transfers etc. Send £5. K Palmer, 59 Park Road, Sheffield S8 7DS.

ACCESSORIES

MAPMAKER

Simply the best way of making adventure maps write on, wipe off. A3 size 420 × 297mm. White PVC screen printed with a blue grid of 11 × 13 boxes. Price includes special pen.
Send ch/PO for **£4.48** to
MAPIT, 166 Robert Street, YNYSYBWL, Mid Glamorgan CF37 3EA.

FOR SALE

GADGET 64 £5.95
This 100% machine code CBM64 cassette utility offers: Append, Delete, Renumber (including Goto, Gosub, etc.) Satisfyingly simple to use. Send your Name, Address & P/O or Cheque for £5.95 (P&P Free) to:
J&J GILMORE LTD,
Dept YC1085, 4 Copse Close, Liss, Hampshire, GU33 7EW. England.
Tel: 0730-893430.

SOFTWARE

COMMODORE 64 DATABASE — DATAFILE

Powerful options include: print all/selective records, delete, amend, save, load, copy, alphanumeric sort, user defined fields, memory check, total numeric fields, multi field interrelational, global and between limits searches, etc. Fully menu driven with comprehensive, easy to use instructions.

CBM 64 (compiled) cassette £7.45
CBM 64 (compiled) disk £10.45

COMMODORE 64 WORDPRO — TEXTFILE

Powerful word processor, facilities include — large text buffer, auto centre, variable tab, insert, amend, delete, copy, move text, word count, right justify, paging, join files, multiple file handling, print all/part text, etc. Fully driven with comprehensive, easy to use instructions. Great value.

CBM 64 (compiled) cassette £8.25
CBM 64 (compiled) disk £10.75

All programs with full documentation and guarantee and can be used with all direct connect CBM 64 printers including 1520 print/plotter. Also most centronics (with sutiable hardware interface). Price includes fast delivery by first class post. Cheques/POs to:

A & C SOFTWARE
Dept. 4, 75 Oakthorpe Gardens, Tividale, West Midlands B69 2LF
Telephone: 021-557 8596 (7-9pm)



PROM-64

EPROM programmer for 8K, 16K, ZIF socket, Vpp generator & software on tape **£34.75**
64-CARTridge, 2 × 8K **£5.95**
DEMO EPROM 8K **£4.95**
DHOBI-2 EPROM eraser **£22.95**
UK + 15% VAT

Cambridge Microelectronics
One Milton Road, Cambridge
CB4 1UY
Tel: (0223) 314814 Tlx: 81574

ACCOUNTS SOFTWARE

CASH TRADERS' BOOK KEEPING

Purchase Day Book.
Daily Takings.
VAT Return (A,B,C,D,F).
Bank Statement.
Petty Cash Statement.
Detail listing of up to 99 nominal a/cs.
19 accounts for creditor control.
Final Accounts.
£199.00 (inc. VAT and postage)
Demonstration Pack: £11.50
(refundable against purchase)
CASH WITH ORDER TO:
Ref: YC, Quick-Count Ltd., 15 Neeld Cres., London NW4 3RP. 01-202 5486
(ans mc.)

CLUBS

LOOK! Free Membership. Exchange your used or unwanted Commodor Software. First snap free! SAE for details to: Softsnap UK (YC) 85, Malmesbury Road, Cheadle Hulme, Cheshire, SK8 7QL.

SOFTWARE APPLICATIONS

HOME ACCOUNTS. Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP inbuilt accuracy check. Records all transactions. Projects cashflow for any period ahead. Available for C.16 CBM64 or VIC-20. £8.45 or *free* details from Discus Software, Freepost, Windmill Hill, Brixham TQ5 9BR. Telephone 08045 55532.

HOME ACCOUNTS

Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP in-built accuracy check. Reocrds all transactions. Projects cash-flow for any period ahead. Available for C.16 CBM64 or VIC-20. £8.45 or *free* details from:

Discus Software
Freepost,
Beach Approach
Brixham TQ5 8BR
Tel: 08045 55532

FOR HIRE

HIRE HIRE

STAFFS COMPUTER LIBRARY

CBM 64 OWNERS

Join the friendly Library. We give you what you want. If a title is not on our list then we will get it, if it's still available. (Games only)

- All the latest Titles
- Full 7 Days Hire
- Low Hire Cost. From 75p
- Up to 4 Games per week
- All games original
- Lots of American Titles
- Low Hard and Software prices
- Membership ONLY £5.00

Free m/c Monitor on joining

Send your cheques/postal orders payable to:
STAFF COMPUTERS
The Newlands, Greensome Lane, Stafford

Attention! Commodore Users.

THERE'S A WHOLE LIBRARY OF SOFTWARE FOR YOU AT THE MIDLAND COMPUTER LIBRARY..

LIFE MEMBERSHIP – JUST £6!

HIRE YOUR FIRST GAME FREE!

- Choose from the latest titles on cassette and disc
- Hire for just £1.25 inc. p+p for 7 days
- Hire as many games as you want
- All games are original with full documentation
- Business and educational software too
- Fast 'return-of-post' service
- Low prices if you want to buy
- 14 day money back guarantee
- Over 1000 games in stock including US imports

MIDLAND COMPUTERS

Please send a cheque or postal order for £6.00 stating your machine, or send a large S.A.E. to:-
Midland Computers Limited,
28 College Street, Worcester WR1 2SL. Telephone (0905) 611072

COMMODORE C16 CASSETTE HIRE

AT LAST!. All the top games available for the C16 can now be hired from:

DABSOFT

Library constantly updated as new games become available. Membership fee of **£6.50** includes 2 FREE rentals!
OVER 50 TITLES
One price for any game. Charge of £1.25 inc P&P BOTH WAYS.
For complete memebership kit post cheque or postal order to:

DABSOFT (YC)
39 Guildford Road, Levenshulme M19 3FG. Tel: 061 445 4375.

VIC-20 AND CMB-64 SOFTWARE HIRE

300 titles. Free membership. Send 2 × 17p stamps for your hirekit to: **VIC-20 SOFTWARE HIRE (Y.M.), 8 Thurland Street, Nottingham.** PLEASE STATE MACHINE.

COMMODORE 64 cassette library. Hire latest games and educational software, only one price £1.50 per cassette 7 days hire. Details S.A.E. please: C.C.L., 19 Darrell Way, Abingdon, OX14 1HQ.

SOFTWARE GAMES

CBM 64

Soccer Supremo, soccer management game (HCW ★★★★★) : Pirate Island, board game simulation (HCW ★★★★★) : Cricket, limited over test match (HCW ★★★★). £4.99 each. 2 for £8.99 or 3 for £12.50 on casette. Add £2.50 to total for disc.
From: Dromeda Software, 56 Wells Street, Haslingdon, Lancashire, BB4 5LS.

TO FILL THIS SPACE RING 01-437 0699

TERMS

CLASSIFIED ADVERTISING TERMS & CONDITIONS

Our terms for new advertisers (semi-display and lineage) are strictly pro-forma payments until satisfactory reference can be taken up (excluding recognised advertising agencies). Cheques PO's should be crossed and made payable to:

ARGUS SPECIALIST PUBLICATIONS LTD.,

and send together with the advertisement to:
THE CLASSIFIED DEPT., L/H, NO: 1 GOLDEN SQUARE, LONDON W1R 3AB.

There are no reimbursements for cancellations. Advertisements arriving too late for a particular issue will be inserted in the following issue unless accompanied by instructions to the contrary.

All advertising sales are subject to Government regulations concerning VAT. Advertisers are responsible for complying with the various legal requirements in force eg: The Trade Description Act, Sex Discrimination Act & The Business Advertisements (Disclosure) Order 1977.

FULL TERMS & CONDITIONS OF ADVERTISING AVAILABLE ON REQUEST

COMMODORE 64

SPEED KING™

SPEEDKING – the motorcycle racing game with thrilling high speed wheel-to-wheel action against 19 other determined riders. Work your way up to world class with the 3 skill levels on 10 world famous circuits. Put on your helmet and enjoy the exhilarating challenge of high performance bike racing at 250 mph!

- 10 world famous circuits
- 3 skill levels
- High performance bike with 6 gears
- 19 other riders
- Display of: best position, fastest time, lap record
- Joystick compatible

BY MERVYN ESTCOURT



Illustration by kind permission of Heron Suzuki Racing.

Cheques payable to Digital Integration Ltd

I enclose a cheque/P.O. for ____________ Total

Name ____________

Address ____________

Or debit my Access Account No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Access Tele Sales (0276) 684959

Please send to Digital Integration, Watchmoor Trade Centre, Watchmoor Road, Camberley, Surrey GU15 3AJ.

VAT and p. & p. inclusive within UK (Overseas inc. 55p. per cassette)

Trade and Export enquiries welcome. Lifetime Guarantee.

COMMODORE 64	
Speed King Disk	☐ £12.95
Speed King Cassette	☐ £9.95
Fighter Pilot Disk	☐ £14.95
Fighter Pilot Cassette	☐ £9.95
SPECTRUM	
Night Gunner 48K	☐ £6.95
Fighter Pilot 48K	☐ £7.95
AMSTRAD CPC 464	
Fighter Pilot	☐ £8.95
Fighter Pilot Disk	☐ £13.95

DIGITAL INTEGRATION

Watchmoor Trade Centre,
Watchmoor Road,
Camberley, Surrey
GU15 3AJ

Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play Space Invaders.

Financial planning is a rather grand name for something you've been doing all your life – making ends meet! Perhaps if Mr Micawber had used **BUSICALC** he would have been able to balance the books a little better.

For home, club or small business use BUSICALC 1 should pay for itself in no time at all; for larger companies we recommend BUSICALC 3, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a powerful musical instrument you need to be a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER!**

To use MUSIC MASTER requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming either! MUSIC MASTER is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

INTERDICTOR PILOT is a space flight simulator. Nowadays simulators are widely used to train pilots and astronauts because – to be frank – it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 22nd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilizations is being fought. A shortage of trained pilots has prompted the Federation to develop a computer simulation that allows raw recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 you too can learn to pilot the Interdictor Mk 3 craft. But be warned – this is no game!

SUPERSOFT

SUPERSOFT, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex HA3 7SJ
Telephone: 01-861 1166

Other SUPERSOFT products include the **MIKRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional (most of our competitors use it!). The **VICTREE** cartridge adds dozens of commands to Basic including toolkit aids and disk commands; or on disk there's **MASTER 64,** a really comprehensive package for the keen programmer.

Of course, we do also publish games programs, and with classics like **STIX, QUINX** and **KAMI-KAZE** in our range we are one of the market leaders. But we most enjoy coming up with the sort of programs that are going to be in use for months and years, not hours and days – the sort of programs that make you glad that you bought a computer – and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket. But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque (pre-paid orders are post free!), by calling at our offices, or over the telephone using your ACCESS card.